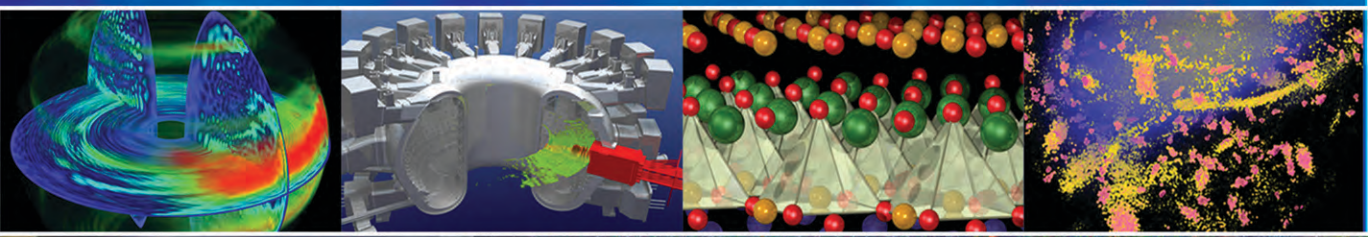
Scientific Discovery through Advanced Computing

U.S. DEPARTMENT OF ENERGY

Office of Science

SciDAC 2010



Conference Proceedings

The Chattanoogan

Chattanooga, Tennessee

July 11 - 15

Recommended citation:

Proceedings of the 2010 Scientific Discovery through Advanced Computing (SciDAC) Conference. Chattanooga, Tennessee, July 11-15, 2010. Oak Ridge National Laboratory. http://computing.ornl.gov/workshops/scidac2010/.

DOCUMENT AVAILABILITY

This report has been reproduced directly from the best available copy. The on-line version is available at http://science.energy.gov/ascr/research/scidac/scidac-conferences/?p=1.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service the trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Copyrights to portions of this report (including graphics) are reserved by original copyright holders or their assignees, and are used by the Government's license and by permission. Requests to use any images must be made to the provider identified in the image credits.

On the cover: Scientific visualizations produced at the Oak Ridge National Laboratory and other laboratories derived from modeling and simulations performed on the U.S. Department of Energy's Jaguar supercomputer and visualized in the Exploratory Visualization Environment for Research in Science and Technology (EVEREST) and other facilities are representative of the nation's investments in addressing grand challenge problems.

SciDAC 2010

Front cover captions

(1) A visualization of the distribution of radio frequency energy in a simulation of the National Spherical Torus Experiment tokamak fusion reactor. Visualization by Sean Ahern

(2) A visualization of the ITER experimental tokamak fusion reactor, including simulated radio frequency energy in the reaction chamber. Visualization by Sean Ahern

(3) The crystal structure of yttrium barium copper oxide, or $YBa_2Cu_3O_7$. Like other members of the cuprate family, this material exhibits superconductivity at high temperatures. For computational simulation, the orange and red copper oxide plane is mapped onto the two-dimensional Hubbard model, and only with recent advances in computing can the predictions of this model be studied systematically. Visualization by Jeremy Meredith

(4) A visualization of a billion-particle simulation of dark matter, identifying features that could lead to galaxy formation. Visualization by Sean Ahern

Back cover captions (left to right)

(1) Accurately simulating the water vapor distribution in the climate system is essential to realistically treat the hydrological cycle and the planetary radiation budget. These images show the simulated monthly-averaged distribution of the total column water vapor from a high-resolution configuration of the CCSM Community Atmospheric Model. Visualization by Jamison Daniel

(2) An atomistic model of cellulose (blue) surrounded by lignin molecules (green) comprising of a total of 3.3 million atoms. Water molecules that are present in the model are not shown. Visualization by Jamison Daniel

(3) The instantaneous net ecosystem exchange of CO_2 is shown projected onto the land surface from a Carbon-Land Model Intercomparison Project (C-LAMP). Green represents an uptake by the biosphere because of photosynthesis where the sun has risen, while red represents a net flux to the atmosphere, dominated by heterotrophic respiration, where it is nighttime. Visualization by Jamison Daniel

(4) A visualization of a simulated thermonuclear flame in a white dwarf wrinkling as it encounters turbulent fuel. At this low density, the turbulent eddies can disrupt the thermal structure of the flame itself, causing a transition to the distributed burning regime.

SciDAC 2010

July 11–15, 2010
Chattanooga, Tennessee, USA

Program Committee

Conference Chair	Thomas Zacharia, Oak Ridge National Laboratory
Accelerators	Panagiotis Spentzouris, Fermi National Accelerator Laboratory
Applied Math	Lori Diachin, Lawrence Livermore National Laboratory
	Petr Plechac, University of Tennessee/Oak Ridge National Laboratory Joint Institute for Computational Sciences
Astrophysics	John Blondin, North Carolina State University
Biology	Jeremy Smith, University of Tennessee/Oak Ridge National Laboratory
Chemistry	Jeff Nichols, Oak Ridge National Laboratory
Climate	Phil Jones, Los Alamos National Laboratory
Combustion	Jackie Chen, Sandia National Laboratories
Computer Science	Kathy Yelick, Lawrence Berkeley National Laboratory
	Bill Gropp, University of Illinois
Data and Data Management	Rob Ross, Argonne National Laboratory
	Terence Critchlow, Pacific Northwest National Laboratory
Exascale Simulation	Thomas Zacharia, Oak Ridge National Laboratory
Facilities	Patricia Kovatch, University of Tennessee/Oak Ridge National Laboratory
	Joint Institute for Computational Sciences
Fusion Sciences	Stephane Ethier, Princeton Plasma Physics Laboratory
Geosciences	Peter Lichtner, Los Alamos National Laboratory
High-Performance Computing	Jack Dongarra, University of Tennessee
Industry	Suzy Tichenor, Oak Ridge National Laboratory
Lattice QCD	Paul Mackenzie, Fermi National Accelerator Laboratory
Materials and Nanoscience	Thomas Schulthess, Swiss Federal Institute of Technology
Nuclear Energy	John Turner, Oak Ridge National Laboratory
Outreach and Tutorials	Rusty Lusk, Argonne National Laboratory
	David Skinner, Andrew Uselton, Lawrence Berkeley National Laboratory
Posters	Jeff Kuehn, Oak Ridge National Laboratory
Program Committee Liaison	Debbie McCoy, Oak Ridge National Laboratory
SciDAC Breakthrough Reports	Tony Mezzacappa, Oak Ridge National Laboratory
Vendor Sessions	Buddy Bland, Oak Ridge National Laboratory
Visualization	Sean Ahern, Oak Ridge National Laboratory
	Ken Joy, University of California, Davis

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6283
managed by
UT-BATTELLE, LLC
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

Contents

Preface xi

Opening remarks

SciDAC-Update xiii
Michael Strayer

Keynote

Computational Challenge in Science, Energy, and National Security xv
Steven Koonin

Materials

Thermodynamics of magnetic systems from first principles: WL-LSMS 1
M. Eisenbach, C.-G. Zhou, D. M. Nicholson, G. Brown, and T. C. Schulthess

Quantum Monte Carlo algorithms: making most of large-scale multi/many-core clusters 7
J. Kim, K. P. Esler, J. McMinis, and D. M. Ceperley

SciDAC breakthroughs

Hybrid quantum simulations of biomolecules: the role of copper in neurodegenerative diseases 12
J. Bernholc, M. Hodak, W. Lu, and F. Rose

Fully 3D multiple beam dynamics processes simulation for the Fermilab Tevatron 23
E. Stern, J. Amundson, P. Spentzouris, and A. Valishev

Applied math

Computational and theoretical analysis of weak and strong transverse-wave structures in gaseous detonations 36
R. Deiterding

Numerical simulation of phase space advection in gyrokinetic models of fusion plasmas 42
M. R. Dorr, R. H. Cohen, P. Colella, M. A. Dorf, J. A. F. Hittinger, and D. F. Martin

Multiple multiresolution representation of functions and calculus for fast computation 53
G. I. Fann, R. J. Harrison, J. Jia, J. Hill, and D. Galindo

Highly parallel solver for multi-scale parquet quantum modeling of strongly correlated materials 63
S. Yang, E. D'Azevedo, H. Fotso, M. Jarrell, J. Liu, T. Maier, C. Sen, and K. Tomko

Parallel mesh control tools for unstructured meshes 69
M. Zhou, A. Ovcharenko, T. Xie, S. Seol, K. E. Jansen, O. Sahni, and M. S. Shephard

Computer science

Verification and validation of current models of thermonuclear-powered supernovae using large-scale simulations 74
Ray Bair, Joe Bernstein, Chris Daley, Benedikt Diemer, Anshu Dubey, Robert Fisher, Jonathan Gallagher, Carlo Graziani, Randy Hudson, George Jordan, Rick Kessler, Yeunjin Kim, Shravan Kumar, Don Lamb, Dongwook Lee, John Norris, Mike Papka, Paul Rich, Katherine Riley, Daan van Rossum, Dean Townsley, Klaus Weide, and Guohua Xia

Extreme-speed scalable direct sparse solvers for heterogeneous supercomputing—An enhancement to the LAPACKrc library 79
Juan Gonzalez and Rafael C. Núñez

MPI at exascale 84
Rajeev Thakur, Pavan Balaji, Darius Buntinas, David Goodell, William Gropp, Torsten Hoefler, Sameer Kumar, Ewing Lusk, and Jesper Larsson Traff

Fusion energy

Coupled whole device simulations of plasma transport in tokamaks with the FACETS code 97
A. H. Hakim, J. R. Cary, J. Candy, J. Cobb, R. H. Cohen, T. Epperly, D. J. Estep, S. Krasheninnikov, A. D. Malony, D. C. McCune, L. McInnes, A. Pankin, S. Balay, J. A. Carlsson, M. R. Fahey, R. J. Groebner, S. E. Kruger, M. Miah, A. Pletzer, S. Shasharina, S. Vadlamani, D. Wade-Stein, T. D. Rognlien, A. Morris, S. Shende, G. W. Hammett, K. Indireshkumar, A. Y. Pigarov, and H. Zhang

Gyrokinetic simulations of edge pedestal turbulence 104
W. Wan, S. E. Parker, Y. Chen, G-Y. Park, C-S. Chang, N. Podhorszki, and S. A. Klasky

Non-diffusive momentum transport and intrinsic rotation in tokamaks 109
W. X. Wang, S. Ethier, S. Klasky, M. Adams, T. S. Hahm, G. Rewoldt, W. W. Lee, W. M. Tang, and P. H. Diamond

Climate

Application of 3-D spherical shell adaptive mesh refinement to an atmospheric model with a vertical Lagrangian coordinate 124
N. G. Andronova, D. R. VandenBerg, Q. F. Stout, R. Oehmke, J. E. Penner, and V. Zubov

Subcycled dynamics in the spectral Community Atmosphere Model version 4 128
M. A. Taylor, K. J. Evans, J. J. Hack, and P. Worley

Earth System Grid Center for Enabling Technologies: Building a Global Infrastructure for Climate Change Research 133
D. N. Williams, J. Ahrens, R. Ananthakrishnan, M. Balman, G. Bell, S. Bharathi, D. Brown, M. Chen, A. L. Chervenak, L. Cinquini, R. Drach, I. T. Foster, P. Fox, S. Hankin, D. Harper, N. Hook, P. Jones, D. E. Middleton, N. Miller, E. Nienhouse, R. Schweitzer, G. Shipman, A. Shoshani, F. Siebenlist, A. Sim, W. G. Strand, F. Wang, C. Ward, P. West, H. Wilcox, N. Wilhelmi, S. Zednik

Combustion

MAESTRO, CASTRO, and SEDONA—Petascale codes for astrophysical applications 144
A. Almgren, J. Bell, D. Kasen, M. Lijewski, A. Nonaka, P. Nugent, C. Rendleman, R. Thomas, and M. Zingale

Simulation of nitrogen emissions in a low swirl burner 149
J. B. Bell, M. S. Day, X. Gao, and M. J. Lijewski

Lattice QCD

Algorithms for Lattice Field Theory at Extreme Scales 154
Richard C. Brower[1], Ronald Babich[1], James Brannick[2], Michael A. Clark[3], Saul Cohen[1], Balint Joo[4], Anthony Kennedy[5], James Osborn[6], Claudio Rebbi[1]

Astrophysics

Evaluating systematic dependencies of Type Ia supernovae 159
A. C. Calder, B. K. Krueger, A. P. Jackson, D. M. Townsley, F. X. Timmes, E. F. Brown, and D.A. Chamulak

Towards exascale supernova simulations with GenASiS 173
C. Y. Cardall, E. Endeve, R. D. Budiardja, P. Marronetti, and A. Mezzacappa

Type Ia supernovae: advances in large scale simulation 183
H. Ma, M. Zingale, S. E. Woosley, A. J. Aspden, J. B. Bell, A. S. Almgren, A. Nonaka, and S. Dong

Visualization

Hybrid parallelism for volume rendering on large, multi-core systems 188
Mark Howison, E. Wes Bethel, and Hank Childs

Performance of communication patterns for extreme-scale analysis and visualization 194
T. Peterka, W. Kendall, D. Goodell, B. Nouanesengsey, H-W. Shen, J. Huang, K. Moreland, R. Thakur, and R. B. Ross

Expressive feature characterization for ultrascale data visualization 199
W. Kendall, M. Glatter, J. Huang, T. Peterka, R. Latham, and R. B. Ross

Geosciences

Cleaning up the cold war: simulating uranium migration at the Hanford 300 Area 205
Glenn Hammond, and Peter Lichtner

Evaluating variable switching and flash methods in modeling carbon sequestration in deep geologic formations using PFLOTRAN 216
C. Lu, P. C. Lichtner, G. E. Hammond, and R. T. Mills

Engineering PFLOTRAN for Scalable Performance on Cray XT and IBM BlueGene Architectures 222
Richard Tran Mills, Vamsi Sripathi, G. (Kumar) Mahinthakumar, Glenn E. Hammond, Peter C. Lichtner, and Barry F. Smith

Simulations of subsurface flow and transport at the pore and continuum scales 228
B. Palmer, Y. Fang, T. Scheibe, and A. Tartakovsky

Task parallel sensitivity analysis and parameter estimation of groundwater simulations through the SALSSA framework 233
Karen Schuchardt, Khushbu Agarwal, Jared Chase, Mark Rockhold, Vicky Freedman, Todd Elsethagen, Tim Scheibe, George Chin, and Chandrika Sivaramakrishnan

Accelerators

Advances in large-scale computations of multi-particle beam dynamics 238
J. Amundson, J. Cary, P. Lebrun, A. Macridin, L. Michelotti, C. S. Park, P. Spentzouris, P. Stoltz, J. Qiang[3] J.-L. Vay, and S. Veitzer

Wakefield computations at extreme-scale for ultra-short bunches using parallel hp-refinement 250
L-Q. Lee, C. Ng, A. Candel, L. Xiao, G. Schussman, L. Ge, Z. Li, V. Rawat, and K. Ko

Dream beams: Extreme-scale computing enabling new accelerator technologies for the energy and intensity frontiers 261
W. B. Mori, W. An, V. K. Decyk, W. Lu, F. S. Tsung, R. A. Fonseca, S. F. Martins, J. Vieira, L. O. Silva, M. Chen, E. Esarey, C. G. R. Geddes, W. P. Leemans, C. B. Schroeder, J.-L. Vay, K. Amyx, D. L. Bruhwiler, J. R. Cary, E. Cormier-Michel, B. Cowan, P. Messmer, C. Huang, and T. Antonsen

Massively parallel simulation of anisotropic Debye shielding in the modulator of a coherent electron cooling system and subsequent amplification in a free electron laser 277
B. T. Schwartz, D. L. Bruhwiler V. N. Litvinenko, S. Reiche G. I. Bell, A. Sobol, G. Wang, and Y. Hao

Recent progress in accelerator physics simulations with the VORPAL code 283
P. H. Stoltz, S. A. Veitzer, C. Nieter, J. R. Cary, K. Amyx, P. Messmer, P. LeBrun, P. Spentzouris, J. F. Amundson, H-J. Kim, and T. Tautges

Advanced software packages for beam dynamics simulation 289
J. Xu, P. N. Ostroumov, P. Fischer, B. Mustapha, M. Min, and J. Nolen

Biology

Modeling the parallel performance of CMA-ES 296
Monte Lunacek, Peter Graf, Ambarish Nag, Christopher H. Chang, Kwiseon Kim, and Michael Seibert

Systems biology approach to discovery of phenotype-specific metabolic processes for engineering bacterial bio-hydrogen 301
A. M. Rocha, W. Hendrix, M. C. Schmidt, J. R. Mihelcic, and N. F. Samatova

Developing improved MD and QMMM codes for modeling enzymes essential to biomass liquid fuel production 306
M. J. Williamson, A. W. Götz, M. K. Garrahan, E. H. Knoll, C. L. Brooks III, R. C. Walker, and M. F. Crowley

High-performance computer simulation in the energy biosciences 312
J. Xu, M. F. Crowley, L. Petridis, H-B. Guo, H. Guo, M. Krishnan, B. Lindner, A. O. Summers, S. M. Miller, L. Liang, J. M. Parks, R. Schulz, M. Saharay, X. Cheng, and J. C. Smith

Data and data management

From grid to cloud, the STAR experience 328
J. Lauret, M. Walker, S. Goasguen, and L. Hajdu

ADIOS: Powering I/O to extreme scale computing 342
Q. Liu, S. Klasky, N. Podhorszki, J. Lofstead, H. Abbasi, C.S. Chang, J. Cummings, D. Dinakar, C. Docan, S. Ethier, R. Grout, T. Kordenbrock, Z. Lin, X. Ma, R. Oldfield, M. Parashar, A. Romosan, N. Samatova, K. Schwan, A. Shoshani, Y. Tian, M. Wolf, W. Yu, F. Zhang, and F. Zheng

Policy-based data management 348
Reagan Moore, Arcot Rajasekar, Mike Wan, Wayne Schroeder, and Chien-Yi Hou

Scientific data management: Challenges and approaches in the extreme scale era 353
Arie Shoshani, Scott Klasky, and Rob Ross

Nuclear energy

Three-dimensional full core power calculations for pressurized water reactors 367
T. M. Evans, G. G. Davidson, and R. N. Slaybaugh

Appendices

Appendix A A-1
Appendix B B-1

Preface

For the past decade, the Scientific Discovery through Advanced Computing (SciDAC) program sponsored by the U.S. Department of Energy's Office of Science has applied emerging capabilities in high-performance computing to compelling scientific and technical challenges. The results of this mindful integration of forefront scientific computing and specialized domain knowledge have been nothing short of extraordinary. SciDAC projects have advanced the development of sustainable energy, improved the understanding of global climate change, accelerated the design of new materials, improved environmental cleanup methods, and expanded the frontiers of physics at scales ranging from the tiniest particles to massive supernovae explosions. They have also been key to the development of a scientific computing infrastructure of breathtaking power.

Perhaps most important, SciDAC has brought together outstanding scientists and mathematicians from national laboratories, universities, and other research institutions, enabling the formation of teams with the capabilities and expertise to tackle the most difficult research challenges. The annual SciDAC conferences have provided the members of these teams and their colleagues in the computational science community a forum for presenting their results, assessing new technologies, and exploring new approaches to collaboration. The SciDAC 2010 Conference, held July 11–15, 2010, in Chattanooga, Tennessee, was no exception.

The SciDAC 2010 program committee assembled an exceptional set of presentations and posters, including an inspiring keynote talk by Steve Koonin, U.S. Department of Energy Under Secretary for Science, on the contributions of extreme-scale computing to science, energy, and national security. The visualization competition provided dramatic and compelling highlights of the range of science supported by the SciDAC program. The fourth SciDAC Tutorials Day, hosted by the University of Tennessee at Chattanooga on July 16, 2010, offered additional opportunities to engage with leading experts and explore new tools and techniques.

It was my great privilege to serve as general chair of SciDAC 2010 and to participate in the celebration of computational science that was a highlight of this conference. I extend my thanks to all of the organizers, program committee, session chairs, presenters, participants, and the conference staff whose hard work was essential to this informative, stimulating, and enjoyable event.

Thomas Zacharia
General Chair, SciDAC 2010
Deputy Director for Science and Technology
Oak Ridge National Laboratory

SciDAC Update
Scientific Discovery through Advanced Computing

Michael Strayer
Associate Director for Advanced Scientific Computing Research
Director, SciDAC (Scientific Discovery through Advanced Computing)
Office of Science, U.S. Department of Energy

Introducing Dr. Steven Koonin as the keynote speaker at SciDAC 2010 and providing an update of SciDAC to open the 2010 conference was Dr. Michael Strayer, associate director of the Advanced Scientific Computing Research (ASCR) program. The Department of Energy's (DOE's) Office of Science oversees ASCR, a program Strayer called "best in class" for advancing science and technology through modeling and simulation. ASCR initiated SciDAC in 2000.

According to Dr. Strayer's slides, ASCR's Recovery Act projects totaled $154.9 million and supported the Advanced Networking Initiative Leadership Computing Facility upgrades, efforts to develop the next generation of computer architectures, meeting the cloud-computing needs of scientists through the Magellan project, and supplementing and leveraging existing SciDAC investments through SciDAC-e, which bolsters DOE electricity grid efforts and the Energy Frontier Research Centers.

Dr. Strayer provided highlights of scientific progress resulting from the Oak Ridge Leadership Computing Facility upgrade to the Jaguar supercomputer, including advances in understanding of energy loss in fusion reactors, properties of next-generation photovoltaic solar-cell materials, and degradation of cellulose for ethanol production. He called out five Gordon Bell winners and three petascale application (DCA++ and WL-LSMS for materials science and NWChem for chemistry). Other scientific highlights included simulations of large, regional earthquakes, next-generation nuclear reactors, magnetic fields that give rise to pulsars, and electronic structures of nanosystems.

In describing the community of computational scientists that SciDAC has built, Strayer spoke of creations of the Centers for Enabling Technologies, SciDAC Institutes, Science Application Partnerships, the SciDAC Outreach Center, the SciDAC Conference, and *SciDAC Review*.

His vision for the future includes computational applied science, science at multiple petaflops, outreach to young investigators, and co-design of hardware and system software as systems approach exascale.

The summary of Dr. Strayer's PowerPoint presentation was prepared by the Oak Ridge National Laboratory. The complete PowerPoint version of his talk at SciDAC 2010 can be viewed at http://computing.ornl.gov/workshops/scidac2010/presentations/m_strayer.pdf.

Computational Challenges in Science, Energy, and National Security

Dr. Steven Koonin
Under Secretary for Science, U.S. Department of Energy

Delivering the keynote speech at SciDAC 2010 was U.S. Department of Energy (DOE) Under Secretary for Science Steven E. Koonin. The title of his talk was "Computational Challenges in Science, Energy, and National Security."

Prior to his confirmation as under secretary by the Senate on May 19, 2009, Dr. Koonin served as chief scientist for BP, where he guided the company's long-range technology strategy, particularly in alternative and renewable energy. He joined BP in 2004 after a 29-year career at the California Institute of Technology as a professor of theoretical physics, including 9 years as provost. His research interests include nuclear astrophysics and theoretical nuclear, computational, and many-body physics, as well as global environmental science. He has advised numerous groups, including the National Science Foundation, the Department of Defense, and the Department of Energy including its national labs.

In his keynote address, Dr. Koonin summarized the importance of collaboration between the SciDAC 2010 attendees. "[Collaboration] brings together these elements I was talking about—laboratory experiments, real-world operating experience, and simulation capabilities or modeling capabilities—to try and make something bigger than each of those bases and have a real impact on the energy program," he said.

Dr. Koonin, who oversees the Office of Science and its programs including Advanced Scientific Computing Research, outlined DOE's missions: sustaining basic research that is discovery-driven and mission-driven; catalyzing transformation of the energy system at national and global levels; enhancing nuclear security; and contributing to U.S. economic competitiveness. Carrying out those missions means addressing such challenges as reducing crude oil use and greenhouse gas emissions.

Simulations accelerate diverse advancements, such as nanostructures for energy storage, next-generation nuclear reactors, and cleaner combustion of fuels in advanced engines and power plants, Dr. Koonin said. Extreme scale computing, with 15 years of world leadership by DOE, can enable simulations that sustain the United States as a major force of innovation and job creation. He cited the example of Cummins's design of a diesel engine solely with computer modeling and analysis; the only testing was after-the-fact to confirm performance. Simulation reduced development time and cost, while achieving a more robust design and better mileage.

Exascale capabilities—roughly one thousand times the speed of today's fastest supercomputers—are expected by 2018, Dr. Koonin said, noting the vigorous pursuit of high-performance computing by other countries.

This summary of Dr. Koonin's PowerPoint presentation was prepared by the Oak Ridge National Laboratory. The complete PowerPoint version of his talk at SciDAC 2010 can be viewed at http://computing.ornl.gov/workshops/scidac2010/presentations/s_koonin.pdf.

Thermodynamics of magnetic systems from first principles: WL-LSMS

M. Eisenbach,[1] C.-G. Zhou,[1*] D. M. Nicholson,[1] G. Brown,[2] T. C. Schulthess[3]

[1]Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA
[2]Department of Physics, Florida State University, Tallahassee, Florida 32306, USA
[3]Institute for Theoretical Physics and Swiss National Supercomputer Center, ETH Zürich, 8903 Zürich, Switzerland

Abstract. Density Functional calculations have proven to be a powerful tool to study the ground state of many materials. For finite temperatures the situation is less ideal and one is often forced to rely on models with parameters either fitted to zero temperature first principles calculations or experimental results. This approach is especially unsatisfactory in inhomogeneous systems, nano particles, or other systems where the model parameters could vary significantly from one site to another. Here we describe a possible solution to this problem by combining classical Monte Carlo calculations—the Wang-Landau method [2] in this case—with a first principles electronic structure calculation, specifically our locally self-consistent multiple scattering code (LSMS) [3]. The combined code shows superb scaling behavior on massively parallel computers. The code sustained 1.836 Petaflop/s on 223,232 cores of the Cray XT5 jaguar system at Oak Ridge.

1. Introduction

Density Functional based first principles electronic structure calculations for condensed matter systems have reached a high level of maturity over the last few decades and are now a standard tool for the study of ground state material properties [1]. While these methods have evolved to provide greater accuracy and deal with wider classes of materials, the field of finite temperature behavior has received less attention. The phase space usually is far too large to be dealt with directly. The usual methods of treating the thermodynamics of a physical system involve either the time evolution of an ensamble or the exploration of the most relevant parts of phase space by means of a Monte-Carlo method. Both these approaches require a large number of evaluations of the underlying Hamiltonian that describes the system ($> O(10^5)$), thus it is usually only feasible to treat severely simplified models that have to be designed to capture the essential physics, as opposed to a direct treatment of the Density-Functional Hamiltonian of the system.

To overcome this limitation we have developed the hybrid Wang-Landau/LSMS (WL-LSMS) code. This code combines revcent advances in computational statistical mechanics, namely the Wang-Landau method [2] with the LSMS first principles method that has already demonstrated superb scalability on massively parallel machines.

[*] Present address: Quantitative Research, J. P. Morgan Chase and Company, New York, New York 10017, USA.

2. Structure of WL-LSMS

The WL-LSMS code uses a multi level parallelization scheme. At the top level, the code parallelizes over concurrent random walkers, where we use a master-slave scheme, with a master that accumulates the density of states of the system, and the slaves that execute the random walks, each running its own instance of the LSMS method. The second parallelization level is the LSMS portion of the code. In typical production runs, the WL method would use a hundred to a few thousand concurrent walkers, and the LSMS portion would be parallelized over up to a few thousand processing cores. The method hence will scale to hundred thousand or millions of processing cores. The schematics of the parallelization structure are shown in Figure 1.

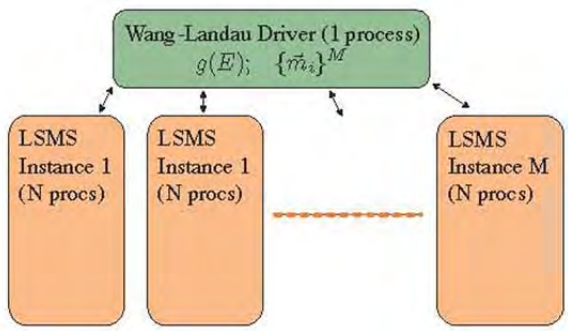


Figure 1. Parallelization strategy of the combined Wang-Landau/LSMS algorithm. The Wang-Landau process generates random spin configurations for M walkers and updates a single density of states $g(E)$. The energies for these N atom systems are calculated by independent LSMS processes (Figure 2). This results in two levels of communication, between the Wang-Landau driver and the LSMS instances, and the internal communication inside the individual LSMS instances spanning N processes each.

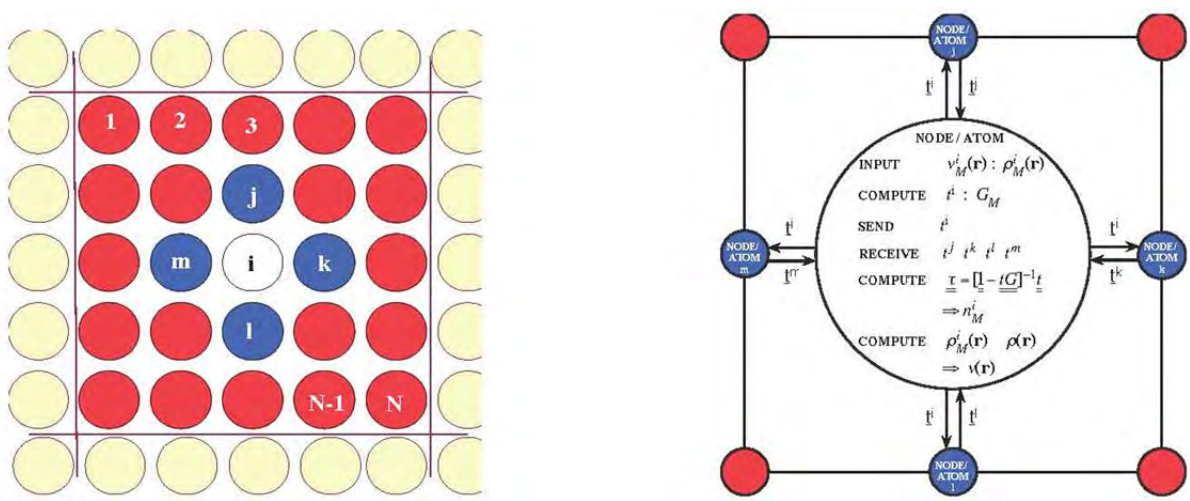


Figure 2. Schematic, Left: LIZ centered at processor/atom i; Right: message passing and computation.

3. The Wang-Landau Algorithm

All thermodynamic potentials can be derived from the partition function

$$Z(T) = \int e^{-E(\boldsymbol{X})/(k_B T)} d\boldsymbol{X} \tag{1}$$

where $E(\boldsymbol{X})$ is the internal energy of the system with the phase space described by the variable $\boldsymbol{X}$ in some high dimensional space consisting of all the microscopic degrees of freedom of the system (atomic positions, velocities, and/or magnetic moments). In importance sampling Monte Carlo simulations one performs a random walk through phase space that is biased in such a way that the walker spends most of the time where the integrand in equation (1) is largest, that is, where the energy $E(\boldsymbol{X})$ is small.

The partition function in equation (1) can be rewritten in the form

$$Z(T) = \int g(E) e^{-E/(k_B T)} dE \tag{2}$$

where the density of states is defined as

$$g(E) = \int \delta(E - E(X)) dX \tag{3}$$

and $\delta(E)$ is the Dirac δ-function.

Flat histogram methods, such as the Wang-Landau algorithm, use the density of states, $g(E)$, for importance sampling, thus accepting the new configuration with probability

$$\min[1, g(E_i)/g(E_{i+1})]. \tag{4}$$

The effect is to create an equal probability of visiting each energy level in the system.

The main obstacle of flat-histogram methods is that $g(E)$ is not known. Instead, an estimate of the density of states $\tilde{g}(E)$ must be constructed self-consistently as the Monte Carlo estimate is generated. The Wang-Landau algorithm accomplishes this as follows. It begins with a prior estimate of the density of states, $\tilde{g}_0(E)$, which might be just a constant. Assuming that a Monte Carlo move to a new configuration with Energy E_{i+1} is accepted according to the criterion of equation (4), the density of states is updated with

$$\ln[\tilde{g}(E_{i+1})] \leftarrow \ln[\tilde{g}(E_{i+1})] + \ln f \tag{5}$$

where f is the modification factor that is initially set to $\ln f = 1$. The modification factor is reduced in steps such that $\ln(f) \leftarrow \ln(f)/2$ and the density of states converges as $\ln(f) \to 0$.

Since the systems we set out to study here have continuous degrees of freedom, $g(E)$ is a function of a continuous variable as well. Hence, we employ the kernel update scheme described by Zhou et al. [8].

4. The LSMS Algorithm

For the energy evaluation, we employ the first principles framework of density functional theory (DFT) in in the local spin density approximation (LSDA). To solve the Kohn-Sham equations arising in this context, we use a real space implementation of the multiple scattering formalism. The details of this method for calculating the Green function and the total ground state energy $E[n(\vec{r}), \vec{m}(\vec{r})]$ are described elsewhere [3,4]. For the present discussion it is important to note that the computationally most intensive part is the calculation of the scattering path matrix τ for each atom in the system by inverting the multiple scattering matrix.

$$\tau = [I - tG_0]^{-1} t$$

The only part of τ that will be required in the subsequent calculation of site diagonal observables (i.e., magnetic moments, charge densities, and total energy) is a small (typically 32×32) diagonal block of this matrix. This will allow us to employ the algorithm described in the next section for maximum utilization of the on node floating point compute capabilities.

Most importantly for the application in the hybrid Wang-Landau LSMS method, our Locally Self-consistent Multiple Scattering (LSMS) method allows the possibility of non-collinear magnetism [5].

The orientation $\hat{e}^i$ of the magnetic moment for each site is determined by

$$\hat{e}^i = \int_{\Omega^i} d\vec{r}\, \vec{m}^i(\vec{r}) / |\int_{\Omega^i} d\vec{r}\, \vec{m}^i(\vec{r})|.$$

Since an arbitrary arrangement is not a DFT ground state we will have to deal with a constrained general state as presented by Stocks et al. [6,7]. In the constrained local moment (CLM) model the LSDA equations are solved subject to a constraint that ensures that the local magnetizations lie along the directions prescribed by $\{\vec{e}_i\}$. Thus this method enables the calculation of the energies of arbitrary orientational states as generated by the Wang-Landau algoritm.

5. Blockinversion in LSMS

The most time consuming part of the LSMS calculation is the inversion of the multiple scattering matrix. The amount of computational effort can be reduced by utilizing the fact that for each local interaction zone only the left upper block (τ_{00}) of the scattering path matrix τ is required. In this section we describe an algorithm that reduces the amount of work needed while providing excellent performance due to its reliance on dense matrix-matrix multiplications that are available in highly optimized form in vendor or third party provided implementations (i.e., ZGEMM in the BLAS library).

The method employed in LSMS to calculate the required block of the inverse relies on the well known expression for writing the invers of a matrix in term of inverses and products of subblocks:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} U & V \\ W & Y \end{pmatrix}$$

where

$$U = (A - BD^{-1}C)^{-1}$$

and similar expressions for V, W, and Y. This this method can be applied multiple times to the subblock U until the desired block τ_{00} of the scattering path matrix is obtained.

6. Performance

We analysed the performance of the code for systems consisting of both 250 and 1024 Fe atoms respectively. For these systems we study the scaling properties of the code as a function of number of walkers used in the WL simulation. Every individual LSMS calculation per walker can be distributed onto 250 or 1024 cores. On the Cray XT5 *jaguarpf* system at ORNL's National Center for Computational Sciences (NCCS), we can thus scale these calculations to up to 895 parallel WL walkers for 250 atoms on 223,752 cores and 218 walkers on 223,232 cores for 1024 Fe atoms. In these performance analysis runs, each walker executes 20 WL steps, which is far fewer than a real simulations.

In Figure 3 we show how the time to solution scales if we increase the number of WL walkers, and thus the total number of samples taken in the WL-LSMS simulation. The result shown thus represent a weak scaling plot, and the scaling behavior of the WL-LSMS method looks close to optimal. We find a similarly optimal strong scaling behavior, if we fix the number of samples taken for every run and increase the number of walkers. With the available size of machines today, we are still far from saturating the method in terms of scaling behavior.

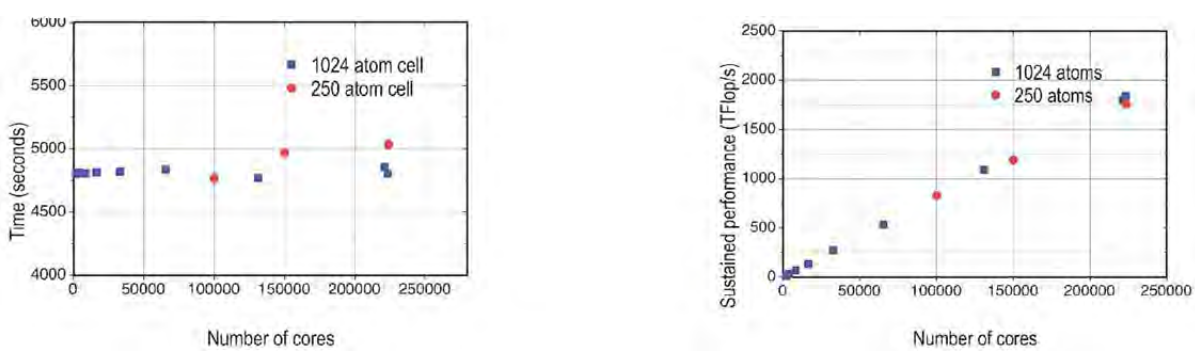


Figure 3. (left) The weak scaling behavior of the WL-LSMS code for 250 and 1024 atom systems with a varying number of Wang-Landau walkers. The times shown represent the total runtime of the code and include the startup costs of the calculations. This accounts for the jump in the runtime for the 250 atom systems as each Wang-Landau walker reads its initial input file. (right) The sustained performance of the Wang-Landau LSMS code on jaguarpf for systems of 250 iron atoms (blue squares) and 1024 atoms (red circles). The code reaches a performance of 1.755 Petaflop/s on 223,752 cores for 250 atoms and 1.835 Petaflop/s on 223,232 cores for a 1024 atom system.

The sustained floating point performance of the runs that correspond are shown also in Figure 3. In order to measure the executed floating point operations of the benchmark runs, we have instrumented the WL-LSMS code with PAPI calls. rFor the largest runs with 218 parallel Wang-Landau walkers of 1024 atoms each and 20 steps per walker, the measured sustained performances was 1.835 petaflop/s, which on the 223,232 AMD Opteron cores running at 2.6 GHz corresponds to a fraction of 79.0% of the theoretical peak performance.

7. Magnetic transition temperature for iron

Here we present a calculation of the magnetic transition temperature (Curie temperature) of iron. Utilizing the methods and algorithm described above we consider a periodically repeated cells of 250 iron atoms and converge the Wang-Landau density of states $g(E)$ for changes in the magnetization direction on the individual iron sites. For the underlying LSMS calculation of these iron cells, the atoms were placed on a body-centered cubic lattice with a lattice parameter of $5.42a_0$, corresponding to the experimental value, and the local interaction zone has a radius of $11.5a_0$. The self-consistently converged potential for the ferromagnetic ground state was used for all the individual frozen-potential energy calculations in the combined Wang-Landau/LSMS algorithm.

Using 400 Wang-Landau walkers the calculation converged in $590{,}000$ and required $4{,}885{,}720$ CPU hours on the jaguar Cray XT5 system. The resulting density of states is shown in Figure 4 and Figure 5. While the computational resources needed for a system with several hundred atoms are considerable, the remaining calculations to compute any desired temperature dependent thermodynamic properties is marginal.

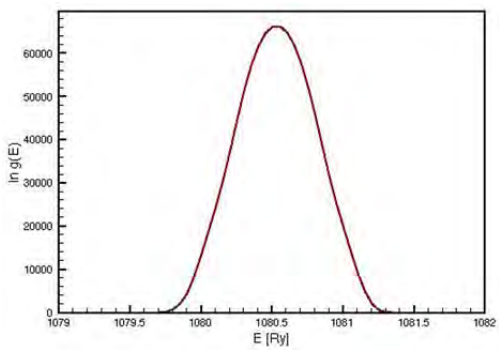


Figure 4. The unnormalized logarithmic Wang-Landau density of states $\ln g(E)$ for a periodic system of 250 iron atoms, respectively.

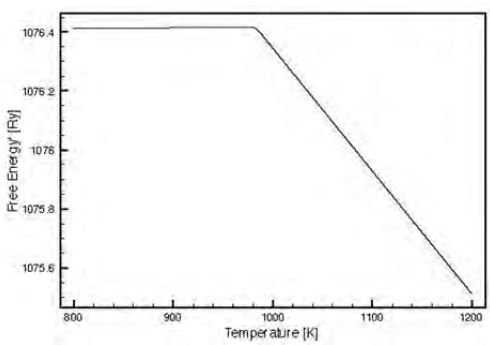


Figure 5. Free Energy F' for a system of 250 iron atoms.

With this density of states the partition function that describes the thermodynamics of the system can be calculated for any temperature.

These calculation yields a transition temperature of 980K. This result is in remarkably good agreement with the experimentally known Curie temperature of bulk iron of 1050K.

8. Summary

In the present paper we have demonstrated our approach for the first principles treatment of finite temperature behavior of magnetic systems. The combination of the most recent massively parallel supercomputing architectures and advances in both algorithmic developments and most importantly new computational methods have made this hybrid statistical mechanics/first principles method feasible.

The code presently is applicable to the evaluation of magnetic transition temperatures of transition metal alloys and has already reproduced the Curie temperature of bulk Iron [9].

Acknowledgments

We acknowledge use of resources of the National Center for Computational Sciences and the Center for Nanophase Materials Sciences, which are sponsored by the respective facilities divisions of the offices of Advanced Scientific Computing Research and Basic Energy Sciences (BES) of the U.S. Department of Energy. The present research was supported by the Laboratory Directed Research and Development fund of ORNL, as well as the division of scientific user facilities of BES. Computations were performed at the Oak Ridge Leadership Computing Facility under an INCITE allocation.

References

[1] Martin, R. M., *Electronic Structure: Basic Theory and Practical Methods*, Cambridge, 2004.
[2] F. Wang, D. P. Landau, *Phys. Rev. Lett.* **86**, 2050 (2001).
[3] Yang Wang, G. M. Stocks, W. A. Shelton, D. M. C. Nicholson, Z. Szotek, and W. M. Temmerman, *Phys. Rev. Lett.* **75** 2867 (1995).
[4] M. Eisenbach, B. L. Györffy, G. M. Stocks, and B. Újfalussy, *Phys. Rev. B* **65**, 144424 (2002).
[5] G. M. Stocks, Y. Wang, D. M. C. Nicholson, W. A. Shelton, Z. Szotek, W. M. Temmerman, B. N. Harmon, V. P. Antropov, *Mater. Res. Soc. Symp. Proc.* **408**, 157 (1996).
[6] G. M. Stocks, B. Újfalussy, X. Wang, D.M.C. Nicholson, W. A. Shelton, Y. Wang, A. Canning, and B. L. Györffy, *Philos. Mag. B* **78**, 665 (1998).
[7] B. Újfalussy, X. Wang, D.M.C. Nicholson, W. A. Shelton, G. M. Stocks, Y. Wang, and B. L. Györffy, *J. Appl. Phys.* **85**, 4824 (1999).
[8] C.-G. Zhou, T. C. Schulthess, S. Torbrügge, D. P. Landau, *Phys. Rev. Lett.* **96**, 120201 (2006).
[9] M. Eisenbach, C.-G. Zhou, D. M. Nicholson, G. Brown, J. Larkin, T. C. Schulthess, SC'09: Proceedings of the Conference on High Performance Computing, Networking, Storage and Analysis, ACM, 2009.

Quantum Monte Carlo algorithms: making most of large-scale multi/many-core clusters

Jeongnim Kim,[1] Kenneth P. Esler,[1] Jeremy McMinis,[2] and David M. Ceperley[1,2]

[1]National Center for Supercomputing Applications
[2]Department of Physics
University of Illinois at Urbana-Champaign, Urbana, IL, USA

Abstract. With advances in algorithms and the changing landscape of high performance computers (HPC), the quantum Monte Carlo method has become a leading contender for high accuracy calculations for the electronic structure of realistic systems. QMC, being statistical, is naturally scalable to a large number of processors. We discuss QMC implementations to overcome the important efficiency and scalability bottlenecks encountered with the HPC systems based on the multi/many-core architecture of today and present state-of-art QMC calculations of solid-state and molecular systems using tens or hundred thousand cores on the petascale computers. Also presented are the solutions for QMC to adapt to the future HPC architectures and to harness ever-increasing computing powers to tackle outstanding materials and chemical problems.

1. Introduction

Continuum quantum Monte Carlo (QMC) methods employ explicitly correlated wavefunctions to describe the many-body effects and symmetries in an efficient and compact manner and solve the Schrödinger equation by a stochastic process [1]. QMC is one of the most accurate ab initio methods to describe the behavior and properties of quantum many-particle systems. It is general and applicable to a wide range of physical and chemical systems in any dimension, boundary conditions etc. The favorable scaling, 3–4 power of the problem size, and ample opportunities of parallelizations, e.g., over multiple Bloch **k**-vectors, make QMC methods uniquely powerful tools to study the electronic structure of realistic systems on large-scale parallel computers.

Until recently, QMC methods have been able to exploit the steady increase in clock rates and enhancement of performance enabled by advances in CMOS VLSI technology for faster and bigger simulations [2]. However, we have seen some significant changes in the mode of increase of computational power in this decade. The performance gain has been largely driven by increasing parallelism in a shared-memory processor (SMP) unit in the form of multi-core processors and GPUs (Graphics Processing Units). Emerging architectures will have an order of magnitude higher concurrency and will force applications to deal with the heterogeneity and hierarchy in the memory, processing and communication subsystems [3]. This presents challenges as well as opportunities for QMC algorithms.

In this article, we discuss the design and implementation details of QMCPACK to take advantage of clusters of multi-core processors and GPUs. QMCPACK, an open-source QMC simulation code written in C++, implements advanced QMC algorithms for large-scale parallel computers [4]. It utilizes hybrid (OpenMP,CUDA)/MPI approaches on multi/many-core architectures to achieve the high computational efficiency, while minimizing communication overhead. We present an overview of applications enabled

by QMCPACK and conclude with the outlook of the QMC algorithms for the current and next generations of HPC systems and our plans for emerging computing architectures.

1.1.QMC Methods

Various quantum Monte Carlo methods solve the time independent Schrödinger equation whose solution, a many-body wavefunction $\Psi(\mathbf{R})$ in a 3N configuration space $\mathbf{R}$, satisfies an eigenvalue equation of the form,

$$\hat{\mathcal{H}}\Psi = E\Psi. \tag{1}$$

Here E is the ground-state energy for the system and $\hat{\mathcal{H}}$ is the *Hamiltonian*, or total energy operator. Since the exact solution Ψ is known only for simple few-body systems, QMC methods employ a trial wave function Ψ_T and find the ground-state solution either by minimizing the energy by varying parameters in a Ψ_T as in variational Monte Carlo (VMC) method or by projecting out the ground state by repeatedly applying an imaginary time Green's function or *propagator* $\exp(-\tau\ \mathcal{H})$ on Ψ_T as in diffusion Monte Carlo (DMC) method [1].

The vast majority of the computational time in an accurate QMC simulation is spent in DMC and therefore minimizing the wall-clock time to obtain a DMC solution within a target error is desired. The optimized computational kernels, physical abstractions and parallelization schemes, however, are also applicable to all other QMC algorithms. We will use DMC to introduce the key concepts and components as implemented in QMCPACK and to guide readers throughout this paper.

In the DMC algorithm, an ensemble of walkers (*population*) is propagated stochastically from generation to generation, where each walker is represented by $\mathbf{R}$. In each propagation step, the walkers are moved through position space by a drift-diffusion process. At the end of each step, each walker may reproduce itself, be killed, or remain unchanged (branching process), depending on the value of the local energy at that point in space, $E_L(\mathbf{R}) = \hat{\mathcal{H}}\Psi_T(\mathbf{R})/\Psi_T(\mathbf{R})$. The walkers in an ensemble are tightly coupled through a trial energy E_T and a feedback mechanism to keep the average population at the target 10^3–10^5 walkers. This leads to a fluctuating population and necessitates shuffling of walkers among parallel processing units to maintain good load balance in an appropriate interval.

1.2. QMC Applications

Figure. 1 highlights recent QMC calculations using the petascale systems at National Center for Computational Sciences (NCCS) and Lincoln GPU cluster at NCSA. Highly scalable and efficient implementations in QMCPACK have allowed QMC studies of the electronic structure of realistic systems with great technological impacts, e.g., on energy, biological and microprocessor development.

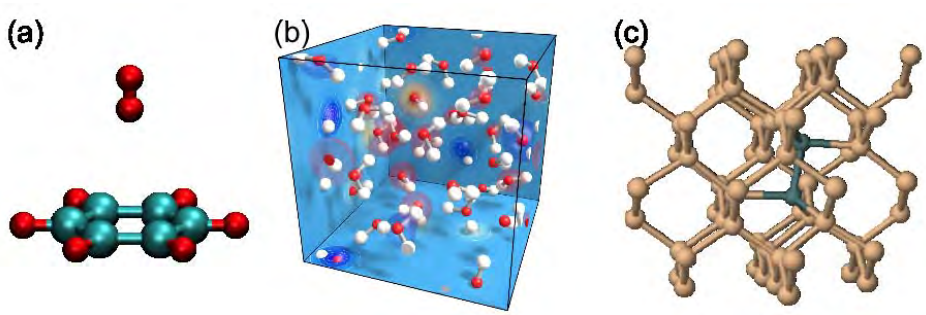


Figure 1. (a) Binding energy of H_2 to a benzene, (b) free-energy of liquid water and (c) energetics of native defects in metals and semiconductors.

2. Hybrid QMC algorithms

Among many ways to parallelize DMC algorithm, the multiplicity of the walkers in an ensemble provides the most natural units for data and task parallelizations. The internal state of each walker, its configuration **R** and other data to reduce recomputing, is encapsulated in Walker class. The operations to propagate each walker during the drift-diffusion process are expressed as a parallel loop over the Walkers. Once a generation has evolved, the properties of all the walkers in an ensemble are collected to determine E_T and N_w, the number of walkers of the next generation, which employs global reduction operations among MPI tasks. The redistribution of Walkers during the load-balance step can be efficiently done by exchanging a serialized Walker object in a large message between paired MPI tasks.

We fully exploit the language features of C++ and standard libraries, e.g., allocators, to maximize cache reuse, eliminate false sharing among threads, reduce the overhead in managing threads and optimize MPI communications. There are several different and independent ways to parallelize a QMC calculation beyond the walker level, e.g., over parameter and configuration spaces and over Bloch **k**-vectors. Such high-level parallelism can be easily managed by mapping a DMC ensemble on a MPI group and is not the subject of this article.

2.1. QMC on Clusters of Multi-Core SMPs

Figure 2(a) illustrates the distribution of Walkers among OpenMP threads and MPI tasks. This hybrid OpenMP/MPI scheme has several advantages over the standard MPI-only scheme.

- Memory optimized: large read-only data to store one-body orbitals and other shared properties to represent the trial wave function and many-body Hamiltonian can be shared among threads, which reduces the memory footprint of a large-scale problem.
- Cache optimized: the data associated with an active Walker are in cache during the compute-intensive drift-diffusion process and the operations on an Walker are optimized for cache reuse. Thread-local objects are used to ensure the data affinity to a thread.
- Load balanced: Walkers in an ensemble are evenly distributed among threads and MPI tasks. The two-level parallelism reduces the population imbalance among MPI tasks and reduces the number of point-to-point communications of large messages (serialized objects) for the Walker exchange.
- Communication optimized: the communication overhead, especially for the collective operations necessary to determine E_T and measure the properties of an ensemble, is significantly lowered by using less MPI tasks.

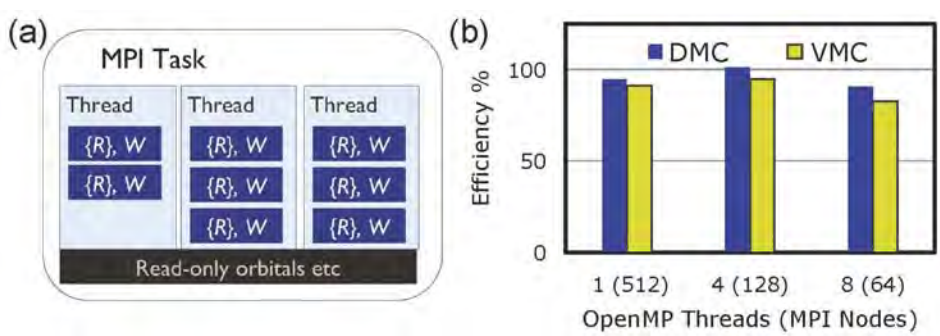


Figure 2. (a) Schematic view of Walker allocation per MPI task. Each Walker object, shown as a box, contains **R**, weight and other data associated with it. (b) Parallel efficiency of OpenMP/MPI hybrid runs on a Cray XT4 (dual quad-core AMD Opteron) with respect to a run using a single core among 8 available cores per node.

Effectiveness of the hybrid scheme is evident in Figure 2(b). The parallel efficiency remains high for any combination of OpenMP threads and MPI tasks even at this modest scale of 512 cores. In fact, DMC

scales nearly perfectly with respect to the number of threads: the additional parallelism from multithreading allows more walkers per MPI task and improves the overall parallel efficiency and load balance among SMP nodes. The VMC efficiency reflects the memory bandwidth-limited nature of QMC algorithms when all the cores are used. However, the performance reduction due to resource sharing on a node, e.g., bandwidth, is insignificant and the added parallelism with more cores improves the efficiency of a QMC simulation by increasing the rate of MC sample generation and, therefore, reducing the time to solution to reach the target error bars of the quantities of interest. The parallel efficiency of hybrid runs is affected by the memory architecture and is subject to the quality of compilers and MPI implementations. In general, the optimal performance is obtained when a MPI task is mapped over a NUMA node as shown in Figure 2(b) with 4 threads on a quad-core processor.

2.2. QMC on Clusters of GPUs

Recently, a path to significant jump in computational performance has become available through the use of GPUs for general-purpose computation. These processors combine many floating point units and relatively simple execution units with a very wide memory bus to allow dramatically higher peak throughput than conventional CPUs. In order to make use of these capabilities, QMC algorithms are reformulated to increase parallelism in the computational kernels: instead of operating on one Walker at a time, which is optimal on cache-based CPUs, we reorder the loops so that each kernel works on all the Walkers in parallel and parallelize the additional loops over the particles or orbitals. Accordingly, the data structures are redesigned to make good use of the memory hierarchy of a GPU and capabilities are added to efficiently manage data transfer between host and GPU. Using mixed precision on GT200 GPUs and MPI for intercommunication and load balancing, we observe typical full-application speedups of 10× to 15× relative to quad-core Xeon CPUs alone, while reproducing the double-precision CPU results within statistical error [5].

2.3. Performance of Hybrid QMC

In Figure 3, we show the calculated speedup for a DMC calculation of a defect in a 64 atom supercell (260 electrons). The parallel efficiency of DMC, which compares the number of MC samples generated in a given wall-clock time, shows near 95% of the ideal speedup up to 216K cores. The scalability of CUDA/MPI is equally high, because the communication overhead can be effectively compensated by increasing the computational load, i.e., by allocating more Walkers on a GPU than a CPU. On average, QMCPACK achieves 25% of the peak performance on x86 systems, which amounts to 0.5 PF sustained performance on jaguar-pf for the 216K-core runs. Our scaling studies expose the performance issues with blocking collectives and parallel I/O at large scales. However, several practical solutions exist including increased intervals between the synchronizations, asynchronous I/O and use of non-blocking collectives [7].

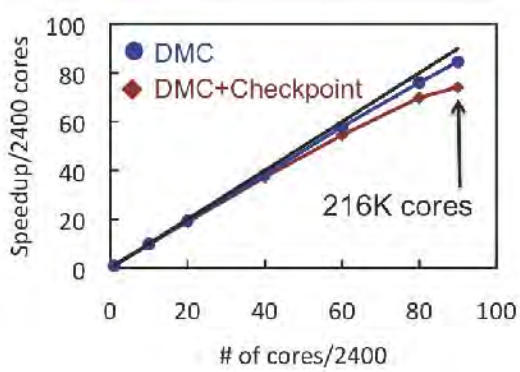


Figure 3. (blue) DMC speedup normalized to 2400 cores. Shown in red is the DMC performance including periodic checkpointing using parallel HDF5 library [6]. The runs use 6 OpenMP threads per MPI task on Cray XT5 (jaguar-pf) systems at NCCS.

3. Conclusions

The quantum Monte Carlo method has exploited the steady increase of computational power over past decades. Increasing parallelism in the form of multi-core SMPs or GPUs is not an exception. We demonstrated that QMC implementations using (OpenMP,CUDA)/MPI hybrid programming model can achieve high scalability and computational efficiency on the current generation of HPC systems. The barriers for sustained petaflop QMC simulations are successfully removed by the hybrid methods which minimize the memory use and communication overhead, while maximizing the use of available parallel processing units on each platform. New developments are in progress to enhance QMC performance by employing light-weight threads, one-sided communications and mixed precision, which will be increasingly critical on the emerging architectures characterized by limited memory per processing unit, heterogeneity, and high concurrency.

Acknowledgments

This work was supported by the U.S. Department of Energy (DOE) under contract No. DOE-DE-FG05-08OR23336 and the U.S. National Science Foundation (NSF) under contract No. 0904572. We used the resources of the National Center for Computational Sciences and the Center for Nanophase Materials Sciences, which are sponsored by the respective facilities divisions of the offices of Advanced Scientific Computing Research and Basic Energy Sciences of DOE under Contract No. DE-AC05-00OR22725. Also utilized are the resources at NCSA and National Institute for Computational Sciences through the NSF Teragrid program under TG-MCA93S030 and TG-MCA07S016.

References

[1] Foulkes W M C, Mitas L, Needs R J, and Rajagopal G. 2001. *Rev. Mod. Phys.* 73 33–83.
[2] Esler K P, Kim J, Ceperley D M, Purwanto W, Walter E J, Krakauer H, Zhang S, Kent P R C, Hennig R G, Umrigar C, Bajdich M, Kolorenc J, Mitas L, and Srinivasan A. 2008. *Journal of Physics: Conference Series* **125** 012057 (15 pp.).
[3] Dongarra J, Beckman P, Aerts P, Cappello F, Lippert T, Matsuoka S, Messina P, Moore T, Stevens R, Trefethen A, and Valero M. 2009. *Int. J. High Perform. Comput. Appl.* **23** 309–322 ISSN 1094-3420.
[4] Jeongnim Kim, K Esler, J McMinis, B Clark, J Gergely, S Chiesa, K Delaney, J Vincent, and D Ceperley. QMCPACK simulation suite URL http://qmcpack.cmscc.org.
[5] Esler K, Kim J, Shulenburger L, and Ceperley D M. 2010. Fully accelerating quantum monte carlo simulations of real materials on gpu clusters submitted to Computing in science and engineering.
[6] Hdf (hierarchical data format) URL http://hdf.ncsa.uiuc,edu/HDF5.
[7] Hoefler T, Lumsdaine A, and Rehm W. 2007. *Proceedings of the 2007 International Conference on High Performance Computing, Networking, Storage and Analysis, SC07* (IEEE Computer Society/ACM).

Hybrid quantum simulations of biomolecules: the role of copper in neurodegenerative diseases

J. Bernholc,[1] M. Hodak, W. Lu, and F. Rose

[1]Center for High Performance Simulation and Department of Physics
North Carolina State University
Raleigh, North Carolina, 27695

E-mail: bernholc@ncsu.edu

Abstract. One of the most important challenges in quantum simulations of biomolecules is efficient and accurate inclusion of the solvent, because the solvent atoms usually far outnumber those of the solute. We have developed a hybrid method that allows for explicit quantum-mechanical treatment of the solvent at low computational cost. It combines Kohn-Sham (KS) density functional theory (DFT) with fixed-density orbital-free (FDOF) DFT. KS DFT is used to describe the biomolecule and its first solvation shells, while the FDOF DFT is employed for the rest of the solvent. The FDOF method scales linearly with the number of atoms and is capable of handling 10^5 solvent molecules on current supercomputers, while taking only a small percentage of the total computational time. The compatibility between the KS and OF DFT methods enables seamless integration between the two. In particular, the flow of solvent molecules across the KS/FDOF interface is allowed and the total energy is conserved. The hybrid method has been used to investigate the binding of copper ions to the prion protein (PrP), which is responsible for a group of neurodegenerative diseases called the transmissible spongiform encephalopathies, such as Cruetzfeldt-Jakob in humans or mad cow disease in cattle. Our results show that as copper ions successively bind to the PrP, the protein changes its shape to accommodate the ions. The binding sequence is such that while the binding energy per copper ion decreases, the energy balance is still positive and up to four copper ions can be bound. PrP acts thus as a copper buffer. Furthermore, the copper-bound PrP becomes more stable and more resistant to misfolding. Therefore, copper plays a beneficial role in the initial stages of prion diseases.

1. Introduction

Accurate representation of solvation is one of the most important aspects of simulations on biomolecules. The solvent not only determines the shape of biomolecules, but it also plays a direct role in many biological processes. The difficulty of dealing with solvation stems from the fact that a large number of solvent molecules is required in a simulated system and the solvent atoms usually outnumber those of the the solute, which are of main interest.

The computational cost of treating the solvent is particularly important for quantum simulations, where systems of $\sim 10^3$ atoms can be treated at present and the cost of the simulation increases dramatically with the number of atoms. For that reason, there is substantial interest in developing methods for efficient description of solvation effects. One way is to use implicit representation of the solvent. In such methods, the solvent is represented as a polarizable continuum dielectric [1,2,3]. These methods are highly efficient, since they eliminate the need for direct treatment of the solvent

molecules. This approach was successfully used [4,5,6,7] in many instances, but evidence suggests that for full understanding of the behavior of many biological systems an explicit treatment is essential. For example, molecular mechanics simulations of proteins have shown that an implicit representation leads to incorrect predictions of native structures and folding pathways, while explicit treatment of the solvent yields results in agreement with experimental observations [8,9]. Furthermore, for many solvated biomolecules water molecules are located near the active sites and are essential to their functions. To properly describe these water molecules, hydrogen bonds with other water molecules have to be included and this can only be done when the solvent is treated explicitly.

Explicit representation of full solvation in quantum calculations is only feasible when the solvent molecules are represented in a simplified manner at a lower level of theory than the solute. A typical example of this approach is the QM/MM method [10,11,12], which combines ab initio and molecular mechanics simulations. In QM/MM, the chemically active part of the system is treated at an ab initio level, while the rest of the system, including the solvent and remainder of the solute, is treated with classical force fields methods. The weak point of QM/MM schemes is the treatment of interaction between the QM and MM subsystems. A special energy term is required to describe this interaction. It usually employs force field parameters, which were developed and tested in calculations within classical molecular mechanics and not for interactions between the QM and MM subsystems. Furthermore, to allow for the flow of solvent molecules across the QM/MM interface, resulting in exchange of molecules between the subsystems, buffer regions and force smoothing functions have to be used [13].

We have recently developed [14] a different approach to hybrid simulations, in which two density functional methods are combined. The chemically active part of the system is treated with the usual ab initio Kohn-Sham DFT, while a frozen-density orbital-free (FDOF) DFT method, which scales linearly with the number of atoms, is used for the rest of the system. In this method, the solvent molecules are assumed to have rigid geometries and frozen electron densities. The advantage of combining two DFT methods is that interactions between the subsystems can be treated at the quantum level rather than at the level of molecular mechanics, as is usually the case in QM/MM methods. In addition, the KS/FDOF method provides seamless integration between the subsystems and allows for the flow of the molecules across the KS/FDOF interface without the need for a buffer region or force smoothing. Within the KS/FDOF method, energy conserving molecular dynamics simulations can be run even when molecules are exchanged between the subsystems.

As an initial application, this method was used [15] to study interactions of copper ions with the prion protein (PrP). PrP is responsible or a group of neurodegenerative diseases called transmissible spongiform encephalopathies (TSEs) [16,17]. The TSEs include bovine spongiform encephalopathy or "mad cow disease" in cattle, scrapie in sheep, chronic wasting disease in deer and elk, kuru and Creutzfeldt-Jakob disease in humans. The PrP structure is characterized by a folded, mostly α-helical, C-terminal domain and an unfolded, flexible, N-terminal region. In TSEs, the prion protein misfolds and aggregates into amyloidic deposits. The detailed structure of the misfolded PrP has not been resolved yet, but it is known that the deposits are β-sheet rich [18]. The normal function of the PrP^C in healthy tissue is still unknown. However, it is now well established [19,20] that the PrP^C efficiently binds Cu^{2+} and further studies [21] link this ability to its function.

To better understand the copper-related function of PrP, copper attachment to PrP has to be fully elucidated. To this end, we use the KS/FDOF hybrid method to study interactions between copper and PrP in various binding modes, which have been suggested experimentally [22]. In particular, we calculate for the first time the attachment geometries and binding energies for low and medium copper concentration conditions. Our analysis of the energetics shows that PrP is capable of carrying out copper buffering function as previously suggested [23]. Starting from these copper-attachment geometries, we perform molecular mechanics simulations of the full length PrPC. These reveal that binding of copper leads to development of new structural features in the unfolded part of the protein increasing the stability of the entire copper-coordinating region. Since the stabilized protein has a

lesser tendency to misfold, this result suggests a protective role of copper in the initial stages of prion diseases.

2. Hybrid Kohn-Sham DFT/Frozen-Density Orbital-Free DFT Method

Within KS/FDOF DFT, the solvent molecules are described with a frozen-density orbital-free (FDOF) DFT method. These molecules are assumed to have rigid geometries and frozen electron densities. The combined total energy is given by the following expression:

$$E_{FDOF}[\varrho] = T_{OF}[\varrho] + E_H[\varrho] + E_{XC}[\varrho] + E_{PPlocal}[\varrho] + E_{AA} \tag{1}$$

where T_{OF} stands for the orbital-free (*OF*) kinetic energy (*KE*) functional, E_H is the Hartree energy, E_{XC} is the exchange correlation energy, $E_{PPlocal}$ evaluates the energy of interaction between the electrons and the local part of the pseudopotentials, and E_{AA} denotes the interaction energy between the atomic cores in the system. Any *OF KE* functional can be used in the formula above, but we have found that the LLP [24] functional works well in calculations for liquid water:

$$T_{LLP} = C_{TF} \int_{\Omega} \varrho^{5/3} \left(1 + \frac{0.0045x^2}{1 + 0.0253x \sinh^{-1} x}\right) d\mathbf{r}, \tag{2}$$

where

$$x = 2^{1/3} \frac{|\nabla \varrho(\mathbf{r})|}{\varrho^{4/3}(\mathbf{r})}. \tag{3}$$

Note that only the local part of the pseudopotential is used in Eq. 1, the reason being that the non-local part of pseudopotential is applied to orbitals rather than the charge density. Neglecting the non-local contributions is well justified, because the present method only deals with intermolecular interactions (molecules are assumed to have frozen geometries and charge densities), and the short-range non-local effects are negligible at typical intermolecular distances.

For expediency, the charge density for each solvent molecule is described by gaussians centered on ionic positions. The coefficients are adjusted so that the charge and the dipole moment of the water molecule are correctly reproduced. The obtained radial distribution functions (RDFs) of this simple model compare well to experimental results for liquid water, as shown in Figure 1.

To define a hybrid calculation scheme combining the FDOF and KS methods, consider a system that contains two types of atoms: those that are treated with the FDOF method (FDOF atoms) and those that are treated with the KS DFT method (KS atoms). The FDOF atoms have fixed charge densities attached to them, which sum up to $\varrho_{FDOF}(\mathbf{r})$, while the charge density corresponding to the KS atoms, $\varrho_{KS}(\mathbf{r})$, minimizes the energy functional for the hybrid system. The total charge density of the entire system is simply the sum of both densities

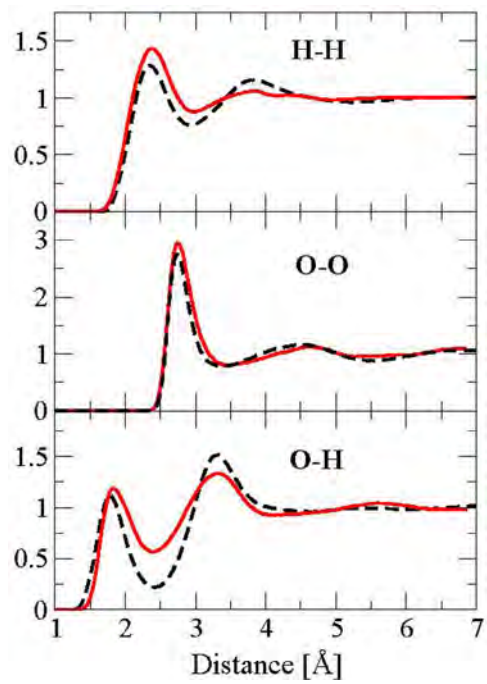


Figure 1. Radial distribution functions calculated using the frozen-density method (solid lines) for a system of 432 water molecules at 300 K. Experimental results [25] are also shown (dashed lines). For better comparison, intramolecular features in H-H and O-H plots were removed from the experimental data.

$$\varrho_{tot} = \varrho_{FDOF} + \varrho_{KS}. \tag{4}$$

Since the Hartree and exchange-correlation energies are explicit functions of the charge density, it is straightforward to calculate these quantities for the hybrid system using ϱ_{tot}. The atom-atom interaction energy is also straightforward, being the sum over all atoms present in the hybrid system, regardless of their type. As for as the pseudopotential energy, the local and non-local parts are treated differently: each atom has a local potential associated with it and it acts on the total density ϱ_{tot}, while only the KS atoms have non-local potentials associated with them and these only act on the KS orbitals.

Defining KE of the hybrid system is somewhat more complicated. Taking the sum of the FDOF and KS KEs is not sufficient, since kinetic energies are non-additive. The non-additive part of KE can be estimated [26] as

$$T_{nadd}[\varrho_{FDOF}, \varrho_{KS}] = T_{OF}[\varrho_{tot}] - T_{OF}[\varrho_{FDOF}] - T_{OF}[\varrho_{KS}]. \tag{5}$$

This gives the following functional for the total electron energy of a hybrid system:

$$\begin{aligned} E_{Hybrid}[\varrho_{FDOF}, \varrho_{KS}, \{\mathbf{R}_i\}_{FDOF}, \{\mathbf{R}_i\}_{KS}] = \quad & T_{TF}[\varrho_{TF}] + T_{KS}[\varrho_{KS}] + T_{nadd}[\varrho_{FDOF}, \varrho_{KS}] \\ + \quad & E_H[\varrho_{tot}] + E_{XC}[\varrho_{tot}] + E_{PP_{nonloc}}[\varrho_{KS}, \{\mathbf{R}_i\}_{KS}] \\ + \quad & E_{PP_{loc}}[\varrho_{tot}, \{\mathbf{R}_i\}] + E_{AA}[\{\mathbf{R}_i\}]. \end{aligned} \tag{6}$$

Here $\{\mathbf{R}_i\}$ denotes all atoms in the system and $\{\mathbf{R}_i\}_{KS}$ stands for atoms treated by the KS DFT. Note that in this functional ϱ_{FDOF} is rigidly tied to the positions of FDOF atoms, while ϱ_{KS} is determined by minimizing the energy functional above.

An important feature of our method is that it allows for for the flow of molecules across the interface between the regions treated by the different methods. Being able to deal with a change in the number of KS atoms is important, since in dynamical calculations the number of solvent molecules in first solvation shells or within the KS region can change. It is also possible that a solvent molecule originally treated by the KS method moves out of the KS region. Such molecule cannot be treated by the KS DFT anymore, since it is required that all KS atoms are inside the KS region, and the number of KS atoms has to be decreased. When such an event occurs, the KS/FDOF simulation can readily continue, since the total electron energy in KS/FDOF system (Eq. 6) can be calculated for any number of KS and FDOF molecules. However, the exchange causes discontinuities in the total electron energy, due to the fact that the electron energy of a solvent molecule is different when calculated with KS or FDOF methods. Nevertheless, these discontinuities can be simply removed by subtracting differences in total energy before and after each exchange. This procedure yields an essentially constant total energy and smooth kinetic energy curves as a function of simulation time, thus enabling energy-conserving quantum molecular dynamics simulations.

3. Copper-PrP Interaction

As an initial application of the KS/FDOF technique, the binding of copper ion to the prion protein, PrP, is investigated. The PrP is implicated in a group of neurodegenerative diseases called transmissible spongiform encephalopathies. These include bovine spongiform encephalopathy (BSE) or “mad cow disease” and the Creutzfeldt-Jakob disease (CJD). The physiological function of the PrP is unknown, but recent investigations [27,28,21,29,30,31] suggest that it may be linked to its ability to bind copper.

A range of experimental techniques, including EPR [32,33,34], CD [35], X-ray crystallography [36] and NMR [37,38] have been used to investigate Cu^{2+} binding to PrP. These studies determined that, at maximum copper occupancy, the full-length prion protein can bind five or six copper ions. Four of these binding sites are located in the octarepeat domain, residues 60-91, which consists of four tandem repeats of the aminoacid sequence PHGGGWGQ. In this case the copper ions are coordinated by a single histidine residue, deprotonated backbone amide nitrogens, and a backbone carboxyl

oxygen. This mode of binding has been confirmed by theoretical studies [14,39] and the copper geometry is displayed in Figure 2.

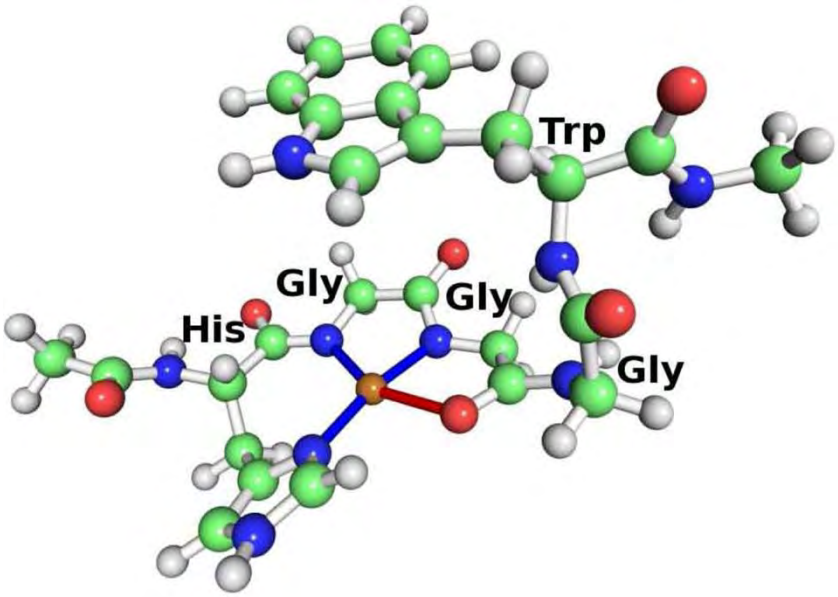


Figure 2. Binding of copper ion to a HGGGW fragment. The copper ion is shown in gold, carbon atoms are colored green, nitrogen atoms blue, oxygen atoms red and hydrogens light grey.

More recent experimental works have found that this picture of copper attachment to the octarepeat domain is incomplete and that different modes of attachment occur depending on environmental factors. EPR investigation by Chattopadhyay et al. [22] demonstrated that copper attachment to the octarepeat region of the prion protein depends on its concentration and that the 1:1 attachment, denoted as Component 1 [22], only takes place at high concentrations. When copper concentration decreases two other coordination modes occur. At low concentration, one copper ion is coordinated by either three or four histidines from different repeats. The exact number of histidines participating in this binding, called Component 3, could not be determined. Interestingly, the same type of binding could be obtained by decreasing pH of the solution [40]. The reason for this is that at low pH, deprotonation of amide nitrogens, which is needed to establish Component 1 binding, becomes unfavorable. The copper ion is thus coordinated by histidine residues instead. At intermediate concentrations, the copper can coordinate to one or two histidines, two water molecules, and a deprotonated backbone nitrogen.

Hybrid KS/FDOF calculations have been used [15] to map out these stages of copper binding. In these calculations, the prion protein was represented by fragments of binding residues to which neutral ends were applied. The fragments, the copper ion and the water molecules closest to the copper ion (6–9 water molecules, depending on the type of binding) were treated by the full Kohn-Sham (KS) DFT. The remainder of the simulation cell was filled with water molecules at liquid density. The water molecules not included in the KS DFT are treated by the orbital-free DFT.

The low concentration binding mode was investigated first. For this type of copper attachment, the experiments found that the copper ion is coordinated by multiple imidazole groups of histidine residues. However, their exact number (three or four) could not be determined. Our calculations consider both these possibilities. Since imidazoles have two nitrogen atoms capable of coordinating metals, commonly denoted as N_δ and N_ε, attachments involving both these nitrogens were investigated for each case. The results show that coordination of the copper ion by four histidine residues through

N_ε atoms is the most favorable energetically. The binding is predominantly planar, with copper-coordinating nitrogens arranged in a slightly distorted square, see Figure 3.

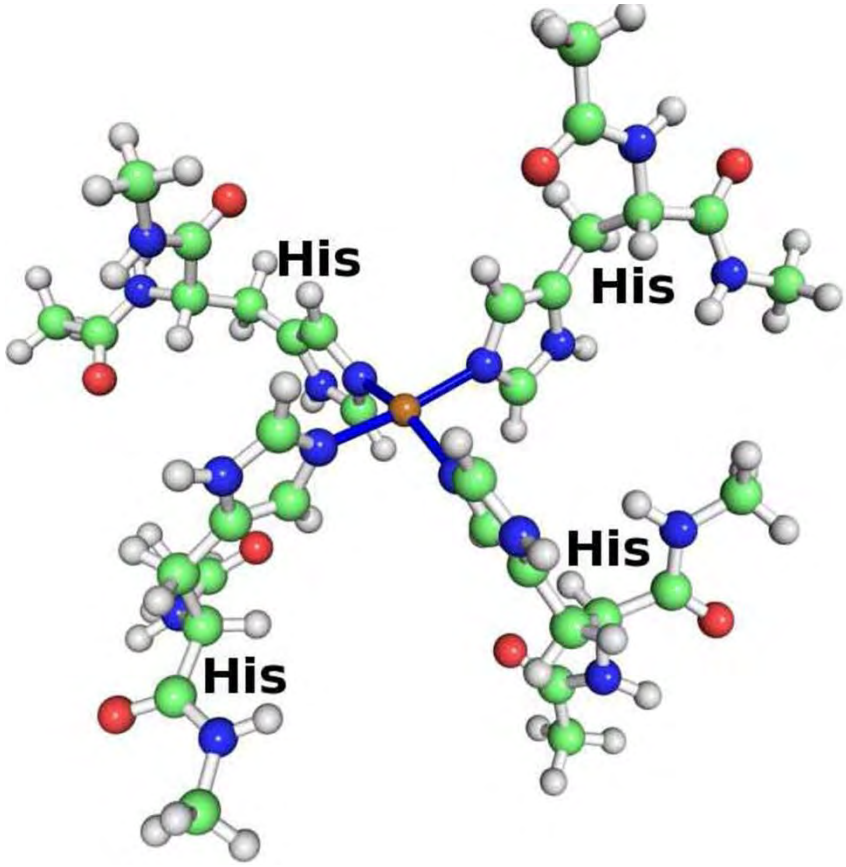


Figure 3. The structure of the low concentration copper binding site in the octarepeat domain of PrP.

For moderate concentration of copper, experiments [22] indicate the involvement of two histidine imidazoles, along in the equatorial and axial positions, although the presence of the axial histidine could not be conclusively proven. Our calculations confirm the presence of axial histidine residues and also show the involvement of two water molecules in the equatorial plane. The obtained binding structure is shown in Figure 4.

4. PrP as a Copper Buffer

One of the most important open questions in PrP research is the normal function of PrP in healthy tissues. After misfolding, it loses the ability to perform this function, which undoubtedly is an important factor in the disease. In fact, several lines of research point to a neuroprotective role of PrP in the brain. For example, comparison between normal and PrP knockout mice shows [41] that the knockout mice have extensive oxidative damage not present in the normal one. Furthermore, neurons containing PrP are more resistant to copper toxicity than cells lacking the protein [27,42,43]. At present, the leading hypothesis for PrP's function in vivo is that it acts as a copper buffer [23], binding uncomplexed extracellular Cu^{2+} and protecting other proteins from its deleterious effects. Using insights from these the copper binding calculations presented above, we now analyze the plausibility of the copper buffer hypothesis.

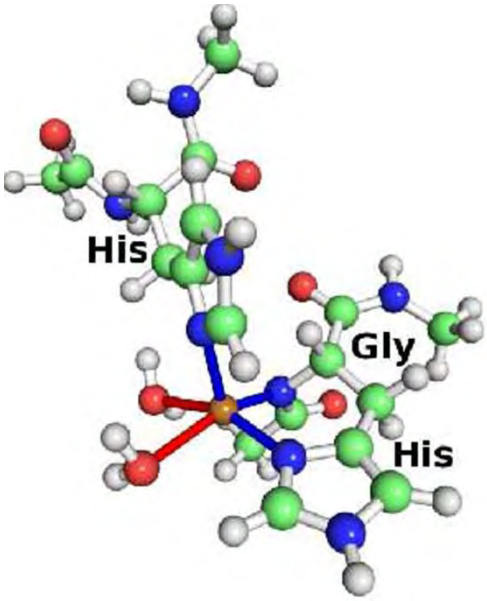


Figure 4. The structure of the medium concentration copper binding mode in the octarepeat domain of PrP.

For each of the three binding modes, we calculate the binding energies of Cu^{2+} to PrP^{C} relative to Cu^{2+} solvated in water. The low concentration, multiple-histidine binding has the energy of 5.22 eV, which is the largest. The medium concentration coordination of copper is less strong, with a binding energy of 2.75 eV. The high occupancy, high concentration binding is the weakest, with an energy gain of 1.76 eV. These results are consistent with recent experimental values for PrP^{C}'s copper affinity [44,45], where the same ordering was found, with low concentration binding being in the nanomolar range, while the high coordination concentration reached micromoles.

For the buffering function of PrP, it is important to consider the number of copper ions that can be coordinated in each of the modes, in addition to the strength of each individual mode. In the low concentration mode, only one copper ion can be coordinated, since all four of the histidine residues in the octarepeat domain participate in its binding. The medium concentration binding mode requires two histidine residues and thus two copper ions can be bound. Finally, four copper ions can be bound to the octarepeat region in the high concentration binding mode, which only requires one histidine residue per each copper ion. The high number of copper ions that can be bound in the high concentration mode means that it is the most energetically favorable overall, even though it is the weakest type of attachment. The energy of 7.04 eV (4 × 1.76 eV) can be gained when four copper ions are bound in this mode. Other binding modes provide smaller energy gains when fully occupied, see Table 1. Note that the binding energy gain increases with copper concentration, confirming that the PrP can act as a copper buffer, since it is energetically favorable to bind more copper ions when they are available in the environment.

Table 1. Energies of copper binding modes for octarepeat domain of PrP

Building mode	Binding energy (eV)	Number of bound ions	Total binding energy (eV)	Free energy change at 300 K (eV)
High concentration	1.76	4	7.04	–2.21
Medium concentration	2.75	2	5.50	–0.44
Low concentration	5.22	1	5.22	0

These results hold true even when the free energies are calculated, because the entropy of the Cu-containing PrP increases when the number of Cu ions bound to PrP increases. This is because the attachment of a single Cu ion, while having the largest binding energy, imposes severe constraints on the internal degrees of freedom of the PrP, bonding to four histidine residues in the octarepeat region. In the intermediate binding mode, only two histidine residues are coordinated to each Cu ion, allowing for increased thermal motion, while in the high concentration case each Cu ion binds to a single residue, leaving the PrP even less constrained. This qualitative argument is confirmed by our classical molecular dynamics simulations on full-length models for all three modes of Cu-bound PrP and calculating entropies using Schlitter's method [46,47]. The results are listed in the last column of 1.

5. Role of Copper in Prion Diseases

In addition to studying local binding geometry of the copper-PrP complex, we also investigate structural properties of full-length copper-bound proteins. The structure of free PrP was also investigated, so that the effect of copper can be determined. Molecular mechanics, as implemented in NAMD [48] is used in this study, due to the large number of atoms required in the simulations. Since the N-terminal domain of PrP^C is unfolded and flexible, the structure of full-length PrP cannot be revealed by current experiments. Even with the use of computer simulation, determining the structure of the full-length PrP^C is challenging, since no experimental data can be used to form an initial structure. Furthermore, because the N-terminal domain is unfolded, even the equilibrium structure is highly flexible, instead of being well defined as in the case of folded proteins. In addition, the unfolded part of the PrP^C takes up significantly more space than a folded protein of the same size and therefore requires a substantially larger simulation cell than a typical folded protein with the same number of residues. To overcome these difficulties, a massively parallel, long-time simulation is used to explore the structural properties of full-length PrP^C.

Snapshots of full-length free and copper bound PrP are shown in Figure 5. Note that the octarepeat region (shown in blue in the Figure 5) has to undergo significant restructuring in order to accommodate all of the binding modes. Such structural flexibility would not be possible for a fully folded protein. Therefore, the flexible character of the N-terminal domain is important, since it allows the PrP to perform its function.

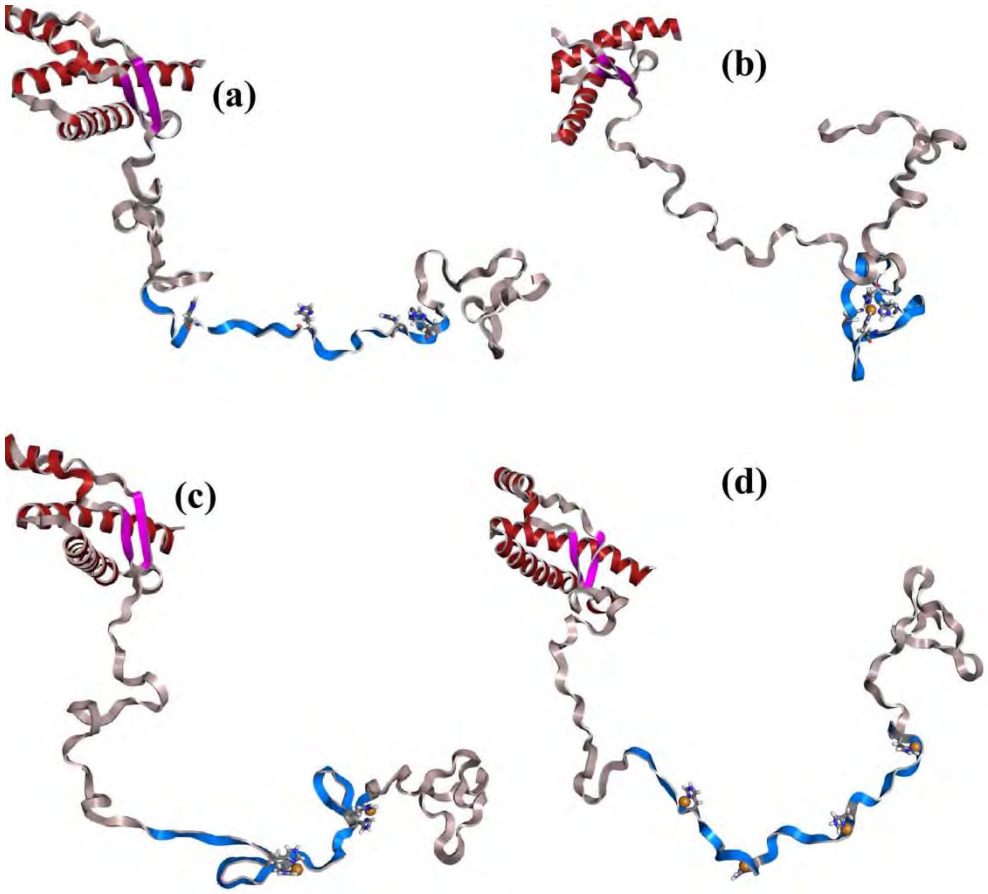


Figure 5. Structures of full-length prion protein in various copper binding modes. (a) Free PrP, (b) Low concentration mode, 1 copper ion attached, (c) medium concentration mode, 2 copper ions attached, (d) high concentration mode, 4 copper ions attached. The octarepeat domain is colored blue.

Since the low concentration binding mode is likely the most physiologically relevant, due to its copper affinity, which is similar to that of other copper-binding proteins [44,49,50], we investigate its structural properties in further detail. A snapshot of its structure is shown in Figure 5b. The binding of copper causes formation of three turns in the intermediate PrP segments connecting the binding histidine residues. Outside of the copper binding domain, the flexible part of the protein behaves in a fashion similar to the free PrP.

Importantly, the simulations also show stiffening of the flexible N-terminal domain upon binding of the copper ion; the unstable structural features formed in the N-terminal domain of PrP persist on average 17% longer in copper-bound PrP, compared to the free PrP. This implies a positive role of copper in the initial stages of prion diseases, since copper-bound PrP is harder to misfold due to stiffening of its structure.

Interestingly, protective effect of copper binding to PrP was also suggested in another computational study performed by Cox et al. [51], even though it considered binding at a different binding site located at His96. This study found that backbone bending at that site is incompatible with the proposed models of misfolded PrP [52,53], thus suggesting a protective role of copper attachment.

6. Summary and Conclusions

Thy hybrid KS/FDOF method provides an effective way for including explicit solvent in quantum biosimulations at low computational cost. In this approach the biomolecule, or its fragment, and solvent molecules in the first solvation shells are treated by full Kohn-Sham DFT, while the remaining solvent molecules are expeditiously described using frozen-density, orbital-free DFT. This method allows for efficient and accurate description of solvent effects and provides seamless integration between regions treated by the KS and FDOF DFT methods.

A first application to multistage copper binding to PrP was also presented. We determined the geometries and binding energies of the low, medium and high concentration modes. The results show that as the number of available copper ions increases, the binding site(s) will spontaneously rearrange to accommodate more copper ions with a net energy gain, despite the fact that the binding strength per copper ion decreases with the number of ions per PrP. The strong affinity for Cu and the multiplicity of concentration-dependent copper binding sites strongly supports the previously suggested copper buffering role of the prion protein as storing excess Cu2+ and protecting other proteins from the effects of uncomplexed copper.

The effect of copper binding on the conformation of the prion protein was also investigated through very large scale molecular dynamics simulation. Our results show that copper binding alters the structure of the unfolded N-terminal domain, creating flexible turns in the octarepeat region that coordinates the copper ion. Furthermore, Cu binding increases the stability of the entire N-terminal domain, increasing its resistance to misfolding. The last results suggest that copper binding in the low concentration mode, which is the dominant one in vivo, delays the onset of prion diseases. The above results demonstrate the usefulness and capabilities of the KS/FDOF approach as a tool for investigating properties of solvated biomolecules and their reactions. This method should be particularly useful for studies of transition metal interactions with solvated biomolecules and of enzymatic reaction mechanisms, where an accurate quantum-mechanical description of fairly large solvated systems is needed.

References

[1] Tomasi J and Persico M. 1994. *Chemical Reviews* **94** 2027–2094.
[2] Rivail J L and Rinald D. 1995. *Liquid state quantum chemistry: computational applications of the polarizable continuum models* (World Scientific, Singapore).
[3] Cramer C J and Truhlar D G. 1996. *Continuum solvation models* (Kluwer, Dordrecht).
[4] Fattebert J and Gygi F. 2002. *Journal of Computational Chemistry* **23** 662–666.
[5] Begue D, Carbonniere P, Barone V, and Pouchan C. 2005. *Chemical Physics Letters* **416** 206–211.

[6] Guimarães L, Avelino de Abreu H, and Duarte H. 2007. *Chemical Physics* **333** 10–17.
[7] Bonifassi P, Ray P, and Leszczynski J. 2006. *Chemical Physics Letters* **431** 321–325.
[8] Zhou R and Berne B. 2002. *Proceedings of the National Academy of Sciences* **99** 12777–12782.
[9] Rhee Y M, Sorin E J, Jayachandran G, Lindahl E, and Pande V S. 2004. *Proc. National Acad. Sciences United States Am.* **101** 6456–6461.
[10] Warshel A and Levitt M. 1976. *J. Mol. Biol.* **102** 227–249.
[11] Singh U C and Kollmann P A. 1986. *J. Comput. Chem.* **7** 718–730.
[12] Field M J, Bash P A, and Karplus M. 1990. *J. Comput. Chem.* **11** 700–733.
[13] Kerdcharoen T and Morokuma K. 2002. *Chem. Phys. Lett.* **355** 257.
[14] Hodak M, Lu W, and Bernholc J. 2008. *J. Chem. Phys.* **128** 014101.
[15] Hodak M, Chisnell R, Lu W, and Bernholc J. 2009. *Proceedings of the National Academy of Sciences* **106** 11576.
[16] Prusiner S. 1997. *Science* **278** 245–251.
[17] Prusiner S. 1998. *Proc Natl Acad Sci USA* **95** 13363–13383.
[18] Baxa U, Cassese T, Kajava A, and Steven A. 2006. *Adv Protein Chem* **73** 125–80.
[19] Hornshaw M P, Mcdermott J R, Candy J M, and Lakey J H. 1995. *Biochem. Bioph. Res. Co.* **214** 993–999.
[20] Hornshaw M P, Mcdermott J R, and Candy J M. 1995. *Biochem. Bioph. Res. Co.* **207** 621–629.
[21] Brown D, Qin K, Herms J, Madlung A, Manson J, Strome R, Fraser P, Kruck T, vonBohlen A, SchulzSchaeffer W, Giese A, Westaway D, and Kretzschmar H. 1997. *Nature* **390** 684–687.
[22] Chattopadhyay M, Walter E, Newell D, Jackson P, Aronoff-Spencer E, Peisach J, Gerfen G, Bennett B, Antholine W, and Millhauser G. 2005. *J. Am. Chem. Soc.* **127** 12647–12656.
[23] Millhauser G. 2007. *Annu. Rev. Phys. Chem* **58** 299–320.
[24] Lee H, Lee C, and Parr R. 1991. *Phys. Rev. A* **44** 768.
[25] Soper A K. 2000. *Chem. Phys.* **258** 121.
[26] Wesolowski T and Warshel A. 1993. *J. Phys. C* **97** 8050.
[27] Vassallo N and Herms J. 2003. *J. Neurochem.* **86** 538–544.
[28] Inestrosa N, Cerpa W, and Varela-Nallar L. 2005. *IUBMB Life* **57** 645.
[29] Millhauser G. 2004. *Acc. Chem. Res* **37** 79–85.
[30] Whittal R, Ball H, Cohen F, Prusiner A, and Baldwin M. 2000. *Protein Sci.* **9** 332–343.
[31] Pauly P and Harris D. 1998. *J. Biol. Chem.* **273** 33107–33110.
[32] Aronoff-Spencer E, Burns C, Avdievich N, Gerfen G, Peisach J, Antholine W, Ball H, Cohen F, Prusiner S, and Millhauser G. 2000. *Biochem.* **39** 13760–13771.
[33] Burns C S, Aronoff-Spencer E, Legname G, Prusiner S B, Antholine W E, Gerfen G J, Peisach J, and Millhauser G L. 2003. *Biochem.* **42** 6794–6803.
[34] Bonomo R, Impellizzeri G, Pappalardo G, Rizzarelli E, and Tabbi G. 2000. *Chem.* **6** 4195–4202.
[35] Viles J, Cohen F, Prusiner S, Goodin D, Wright P, and Dyson H. 1999. *Proc. Natl. Acad. Sci. U.S.A.* **96** 2042–2047.
[36] Burns C S, Aronoff-Spencer E, Dunham C M, Lario P, Avdievich N I, Antholine W E, Olmstead M M, Vrielink A, Gerfen G J, Peisach J, Scott W G, and Millhauser G L. 2002. *Biochem.* **41** 3991–4001.
[37] Jackson G, Murray I, Hosszu L, Gibbs N, Waltho J, Clarke A, and Collinge J. 2001. *Proc. Natl. Acad. Sci. U.S.A.* **98** 8531–8535.
[38] Jones C E, Klewpatinond M, Abdelraheim S R, Brown D R, and Viles J H. 2005. *J. Mol. Biol.* **346** 1393–1407.
[39] Pushie M and Rauk A. 2003. *J. Biol. Inorg. Chem.* **8** 53–65.
[40] Wells M A, Jackson G S, Jones S, Hosszu L P, Craven C J, Clarke A R, Collinge J, and Waltho J P. 2006. *Biochem. J.* **399** 435–444.
[41] Klamt F, Dal-Pizzol F, da Frota M L C, Walz R, Andrades M E, da Silva E G, Brentani R R, Izquierdo I, and Moreira J C F. 2001. *Free Radical Bio. Med.* **30** 1137–1144.

[42] Rachidi W, Vilette D, Guiraud P, Arlotto M, Riondel J, Laude H, Lehmann S, and Favier A. 2003. *J. Bio. Chem.* **278** 9064–9072.
[43] Zeng F N, Watt N T, Walmsley A R, and Hooper N M. 2003. *J. Neurochem.* **84** 480–490.
[44] Wells M, Jelinska C, Hosszu L, Craven C, Clarke A, Collinge J, Waltho J, and Jackson G. 2006. *Biochem. J.* **400** 501–510.
[45] Walter E D, Chattopadhyay M, and Millhauser G L. 2006. *Biochem.* **45** 13083–13092.
[46] Schlitter J. 1993. *Chem. Phys. Lett.* **215**.
[47] Glykos N. 2006. *Journal of computational chemistry* **27** 1765.
[48] Phillips J C, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel R D, Kale L, and Schulten K. 2005. *J. Comput. Chem.* **26** 1781–1802.
[49] Linder M. 1991. *Biochem. Copp.* 73–134.
[50] Masuoka J, Hegenauer J, Van Dyke B, and Saltman P. 1993. *J. Bio. Chem.* **268** 21533–21537.
[51] Cox D, Pan J, and Singh R. 2006. *Biophysical journal* **91** 11–13.
[52] Govaerts C, Wille H, Prusiner S, and Cohen F. 2004. *Proceedings of the National Academy of Sciences of the United States of America* **101** 8342.
[53] Yang S, Levine H, Onuchic J, and Cox D. 2005. *The FASEB Journal* **19** 1778.

Fully 3D multiple beam dynamics processes simulation for the Fermilab Tevatron

E. Stern[1], J. Amundson, P. Spentzouris, and A. Valishev

[1]Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL, USA

E-mail: egstern@fnal.gov

Abstract. The Fermilab Tevatron has been, until 2010, the premier high-energy physics collider in the world. The data collected over the last decade by high-energy physics experiments running at the Tevatron have been analyzed to make important measurements in fundamental areas such as B meson masses and flavor oscillation, searches for the Higgs boson, and supersymmetry. Collecting these data at the limits of detectability has required the Tevatron to operate reliably at high beam intensities to maximize the number of collisions to analyze. This impressive achievement has been assisted by the use of HPC resources and software provided through the SciDAC program. This paper describes the enhancements to the BeamBeam3d code to realistically simulate the Tevatron, the validation of these simulations, and the improvement in equipment reliability and personal safety achieved with the aid of simulations.

1. Introduction

The task of asking and answering the most basic fundamental questions about the nature of time, space, matter and energy falls to the sciences of particle physics and astrophysics. The particle accelerator [1] is the main tool used by experimental particle physicists to create new forms of matter which are studied to understand their interactions and relationship with other well-understood matter. In pursuing new realms of interactions and new forms of matter, physicists use accelerators running at the highest possible energies. Running the accelerator at larger energies allows the exploration of previously unknown regimes through the production of more massive particles. Since the occurrence of new physics processes is likely to be rare, the accelerator has to generate interactions at a high rate so that sufficient numbers of new particles may be produced and identified. In accelerator physics, this is referred to as running with high luminosity.

For the last decade until 2010, the Tevatron operating at the Fermi National Accelerator Laboratory has been the highest energy accelerator in the world. The Fermilab Tevatron [2] is a collider accelerator in which beams of protons and antiprotons circulate in opposite directions and collide at a center-of-mass energy of 1.96 TeV.

Accelerators are large complicated devices. Throughout physical length of the Tevatron, (see Figure 1) which exceeds six kilometers, thousands of magnets for steering and focusing charged particles, radio-frequency cavities for acceleration and bunch shaping, along with the ancillary support equipment including vacuum, cryogenic, power, cooling and diagnostic equipment are assembled and work together in the delicate task of capturing and accelerating beams made of 72 bunches of over 10^{13} mutually repelling charged particles, bringing them into collision while maintaining the beams with minimal particle losses for hours at a time. Beams in accelerators are subject to dynamic physical

Figure 1. View inside the Tevatron tunnel

processes which disrupt normal operations by causing instability and beam loss. The strength of these disruptive effects increases within creasing beam intensity, which interferes with the goal of running high luminosity beams for physics.

Each colliding beam consists of 36 bunches of particles shown schematically in Figure 2. The proton and antiproton beams circulate in opposite directions around the ring moving in a common vacuum pipe. For high-energy physics operations, the beams collide head-on at the B0 and D0 interaction points (IPs) which are surrounded by particle detectors. In the intervening arcs the beams are separated by means of electrostatic separators; long-range (also referred to as parasitic) collisions occur at 136 other locations. Head-on and long-range interactions couple all of the bunches to each other, allowing the possibility of the development of coherent modes of oscillation. The magnetic optics of the horizontal and vertical planes of the Tevatron are strongly coupled, so perturbations in one plane will appear in the other plane. The charged beams also interact electromagnetically with the conducting walls of the vacuum chamber, introducing an impedance effect which distorts and perturbs the bunches. The focusing strength of the magnetic elements in the beamline varies with particle momentum, an effect called chromaticity, which can contribute to, or dampen, the development of impedance driven instabilities [9]. The combination of impedance and beam-beam interactions could also lead to the development of dipole or quadrupole mode instabilities [5].

The combination of all these effects imposes serious limitations on machine operations, hence constant efforts are being exerted to better understand the beam dynamics. Due to the extreme complexity of the problem, a numerical simulation appears to be one of the most reliable ways to study performance of the system. A simulation that takes into account the number of bunches (72), the number of macroparticles needed for a faithful rendering of bunch characteristics (>500000), and the number of interaction regions (~100) can only be performed with high performance computing using optimized algorithms for calculating collective electromagnetic effects amongst many charged particles.

This paper presents a macroparticle simulation that includes the main features essential for studying the coherent motion of bunches in a collider: a self-consistent 3D Poisson solver for beam-beam force computation, multiple bunch tracking with the complete account of sequence and location of long-range and head-on collision points, and a machine model including our measurement based understanding of the coupled linear optics, chromaticity, and resistive wall impedance.

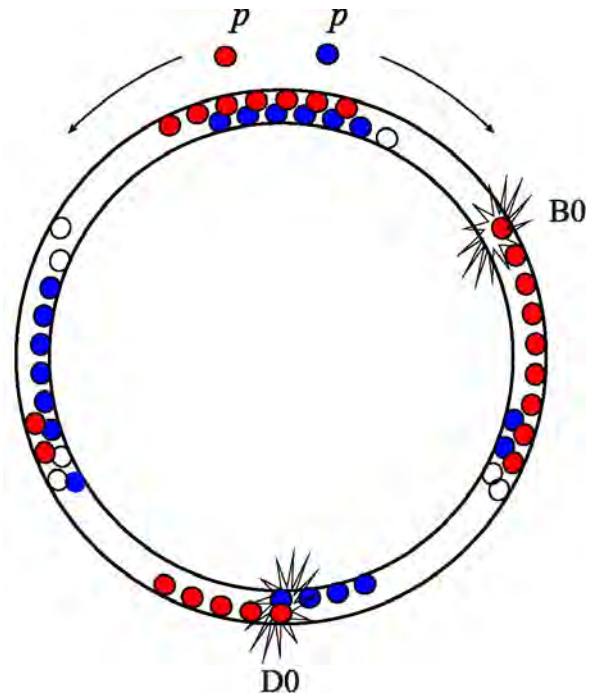


Figure 2. Schematic of the position of proton and antiproton bunches in the Tevatron with 36 proton and 36 antiproton bunches. The diagram shows the positions at a time when the lead bunch of the trains are at the head-on collision location. Head-on collisions occur at location B0 and D0.

We validate individual physical processes against measured data where possible, and analytic calculations elsewhere. Finally, we present simulations of the effects of increasing beam intensity with single and multiple bunches, and study the combined effect of long-range beam-beam interactions and transverse impedance. The results of the simulations were successfully used in Tevatron operations to support a change of chromaticity during the transition to collider mode optics leading to a factor of two decrease in proton losses. This allowed the Tevatron to run with higher intensity while remaining within radiation safety margins, improving reliability of collider operations and the physics discovery potential of the machine.

2. BeamBeam3d code

The Poisson solver in the BeamBeam3d code is described in Ref. [6]. Two beams are simulated with macroparticles generated with a random distribution in phase space. The accelerator ring is conceptually divided into arcs with potential interaction points at the ends of the arcs. The optics of each arc is modeled with a 6 × 6 linear map that transforms the phase space $\{x, x', y, y', z, \delta\}$ coordinates of each macroparticle from one end of the arc to the other. There is significant coupling between the horizontal and vertical transverse coordinates in the Tevatron. For our Tevatron simulations, the maps were calculated using coupled lattice functions [7] obtained by fitting a model [8] of beam element configuration to beam position measurements. The longitudinal portion of the map produces synchrotron motion among the longitudinal coordinates with the frequency of the synchrotron tune. Chromaticity results in an additional momentum-dependent phase advance $\delta\mu_{x(y)} = \mu_0 C_{x(y)} \Delta p/p$, where $C_{x(y)}$ is the normalized chromaticity for x (or y) and μ_0 is the design phase advance for the arc. This is a generalization of the definition of chromaticity to apply to an arc, and reduces to the normalized chromaticity $(\Delta\nu/\nu)/(\Delta p/p)$ when the arc encompasses the whole ring. The additional

phase advance is applied to each particle in the decoupled coordinate basis so that symplecticity is preserved.

The Tevatron includes electrostatic separators to generate a helical trajectory for the oppositely charged beams. The mean beam offset at the IP is included in the Poisson field solver calculation. Different particle bunches are individually tracked through the accelerator. They interact with each other in the pattern and locations that they would have in the actual collider.

The impedance model applies a momentum kick to the particles generated by the dipole component of resistive wall wakefields [9]. Each beam bunch is divided longitudinally into slices containing approximately equal numbers of particles. As each bunch is transported through an arc, particles in slice *i* receive a transverse kick from the resistive wall wake field induced by the dipole moment of the particles in forward slice *j*:

$$\frac{\Delta\vec{p}_\perp}{p} = \frac{2}{\pi b^3}\sqrt{\frac{4\pi\epsilon_0 c}{\sigma}}\,\frac{N_j r_0 < \vec{r}_j >}{\beta\gamma}\,\frac{L}{\sqrt{z_{ij}}} \qquad (1)$$

[zjs1]

The length of the arc is L, N_j is the number of particles in slice j, r_0 is the classical electromagnetic radius of the beam particle $e^2/4\pi\epsilon_0\, m_0c^2$, z_{ij} is the longitudinal distance between the particle in slice i that suffers the wakefield kick and slice j that induces the wake. $\vec{r}_j$ is the mean transverse position of particles in slice j, b is the pipe radius, c is the speed of light, σ is the conductivity of the beam pipe and $\beta\gamma$ are Lorentz factors of the beam. Quantities with units are specified in the MKSA system.

3. Synchrobetatron Comparisons

We will assess the validity of the beam-beam calculation by comparing simulated synchrobetatron mode tunes with a measurement [10] performed at the VEPP-2M 500MeV e^+e^- collider. These modes are an unambiguous marker of beam-beam interactions and provide a sensitive tool for evaluating calculational models. These modes arise in a colliding beam accelerator where the longitudinal bunch length and the transverse beta function are of comparable size. Particles at different z positions within a bunch are coupled through the electromagnetic interaction with the opposing beam leading to the development of coherent synchrobetatron modes. The tune shifts for different modes have a characteristic evolution with beam-beam parameter $\xi = Nr_0/4\pi\gamma\epsilon$, in which N is the number of particles, r_0 is the classical electromagnetic radius of the beam particle, and ϵ is the unnormalized one-sigma beam emittance.

There are two coherent transverse modes in the case of simple beam-beam collisions between equal intensity beams without synchrotron motion: the σ mode where the two beams oscillate with the same phase, and the π mode where the two beams oscillate with opposite phases [12]. Without synchrotron motion, the σ mode has the same tune as unperturbed betatron motion while the π mode frequency is offset by $K\xi$, where the parameter K is approximately equal to, but greater than 1 and depends on the transverse shape of the beams [11]. The presence of synchrotron motion introduces a more complicated spectrum of modes which are denoted by $n\sigma$ and $n\pi$ where n is the number of oscillation fixed-points along the bunch length induced by coupling.

We simulated the VEPP-2M collider using Courant-Snyder uncoupled maps. The horizontal emittance in the VEPP-2M beam is much larger than the vertical emittance. The bunch length (4 cm) is comparable to β_y^* = 6 cm so we expect to see synchrobetatron modes. In order to excite synchrobetatron modes, we set an initial y offset of one beam sigma approximately matching the experimental conditions.

Simulation runs with a range of beam intensities corresponding to beam-beam parameters of up to 0.015 were performed, in effect mimicking the experimental procedure described in Ref. [10]. For each simulation run, mode peaks were extracted from the Fourier transform of the mean bunch vertical position. The simulations were performed with 10^5 macroparticles per bunch. Doubling the number of macroparticles did not change the location or appearance of any mode peaks. In Figure 3, we plot the

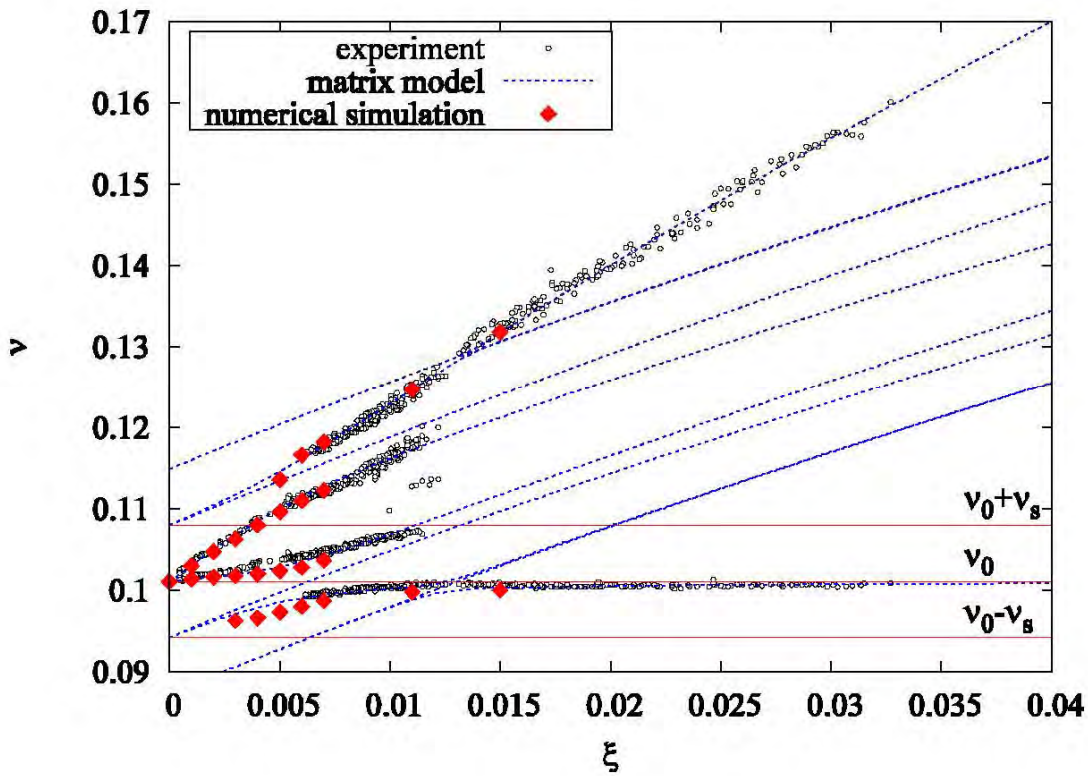


Figure 3. The points show the tunes of experimentally observed synchrobetatron modes at the VEPP-2M accelerator. The red diamonds show the tunes of simulated synchrobetatron modes as a function of beam-beam parameter ξ. The dashed lines show the evolution of mode frequencies predicted by the matrix model as described in the text.

mode peaks from the BeamBeam3d simulation as a function of ξ as red diamonds overlaid on experimental data from Ref. [10] and a phenomonelogical model using linearized coupled modes referred to as the matrix model described in Refs. [13, 14]. As can be seen, there is good agreement between the observation and simulation giving us confidence in the beam-beam calculation.

4. Impedance Tests

Wakefields or, equivalently, impedance in an accelerator with a conducting vacuum pipe gives rise to well known instabilities. Our aim in this section is to demonstrate that the wakefield model in BeamBeam3d quantitatively reproduced these theoretically and experimentally well understood phenomena. Two instabilities considered are the strong and weak head-tail instability analyzed by Chao [15, 9]. These instabilities arise in extended length bunches in the presence of wakefields, synchrotron motion and betatron motion. The wakefield induced by the head of the bunch applies a driving force to the tail of the bunch that couples longitudinal motion to transverse motion. The coupling splits the beam oscillation frequency into upper and lower sidebands at frequencies $\nu_\beta + \nu_s$ and $\nu_\beta - \nu_s$. When he driving force is in-phase with the bunch's normal oscillation, the beam oscillations grow without damping.

The condition for the strong head-tail instability is that the strength of the in-phase driving force (determined by beam intensity) overcomes the stabilizing effect of synchrotron motion. For any particular accelerator optical and geometric parameters, there is an intensity threshold above which the beam becomes unstable. As the beam intensity is increased, the two sidebands move closer to each other. At the point where they meet, the beam becomes unstable.

In Figure 4, we show the evolution of the two modes as a function of beam intensity. With the tune and beam environment parameters of this simulation, Chao's two particle model predicts instability development at intensities of about 9×10^{12} particles, which is close to where the upper and lower modes meet. The model is derived assuming just a leading and trailing particle, while the simulation is done with the full particle distribution, altering the apparent strength of the wake function.

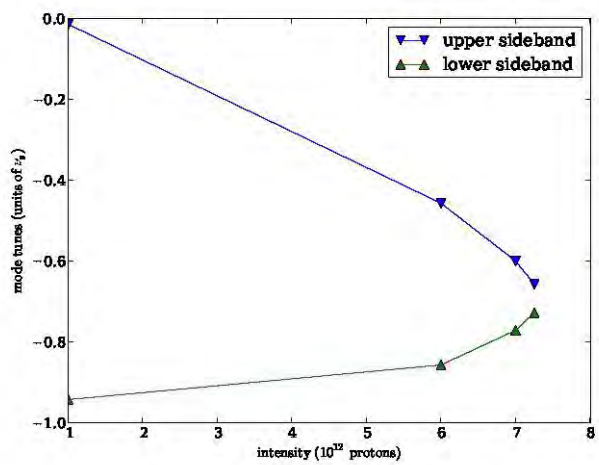


Figure 4. Evolution of the sideband mode frequencies as a function of beam intensity showing the two modes approaching a common frequency due to impedance. The y scale is in units of the synchrotron tune.

The interaction between impedance and chromaticity also causes a head-tail instability characterized by a parameter called the head-tail phase $\chi = 2\pi C \nu_\beta \hat{z}/c\eta$, where η is the slip factor of the machine and $\hat{z}$ is roughly the bunch length. The head-tail phase gives the size of betatron phase variation due to chromatic effects over the length of the bunch. Analysis by Chao [9] predicts a growth rate of

$$\tau^{-1} = \frac{N r_0 W_0}{2\pi \beta \gamma \nu_\beta} \chi \tag{2}$$

where N is the number of particles in the bunch, W_0 is the value of the wake function, β and γ are Lorentz factors and ν_β is the betatron tune and χ is the head-tail phase previously defined. Eqn. 2 predicts a growth rate that varies linearly with χ when the growth rate is normalized by the first factor, which includes all the beam intensity and geometric effects. The more detailed Vlasov analysis [16] also predicts a universal dependence of normalized growth rate versus head-tail phase that begins linearly with head-tail phase [16] and peaks around –1[1].

[1]The simulated machine is above transition (η is positive.) The head-tail instability develops when chromaticity is negative, thus the head-tail phase is negative.

We simulated three different beam intensities with a range of chromaticites from −.001 to −0.5 to get head-tail phases in the 0 to −1 range. We extracted the growth rate from a logarithmic graph of the simulated beam size versus turn number and plot the normalized growth rate vs. head-tail phase in Figure 5. The normalized curves are nearly identical and peak close to head-tail phase of unity. The deviation from a universal curve is again due to differences between the idealized model and detailed simulation.

We conclude that our simulation of resistive wall impedance is valid.

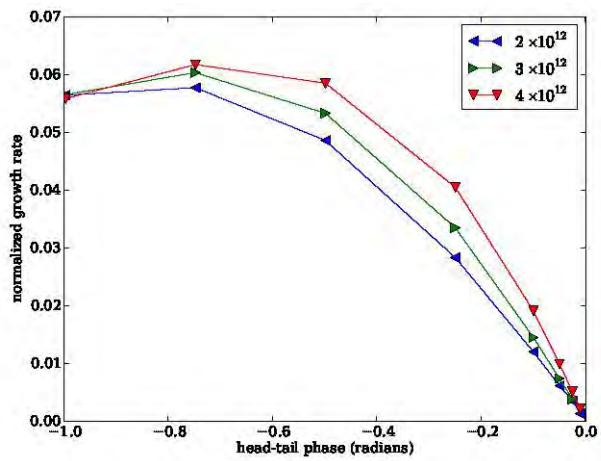


Figure 5. The normalized growth rate of dipole motion in the simulated accelerator with impedance and chromaticity as a function of head-tail phase χ at three beam intensities demonstrating their linear relation-ship close to 0 and the near-universal relationship for head-tail phase between −1 and 0.

5. Bunch-by-Bunch Emittance Growth at the Tevatron

Referring back to Figure 2, we see that there are three trains of twelve bunches for each species. A train occupies approximately 81.5° separated by a gap of about 38.5°. The bunch train and gap are replicated three times to fill the ring. Bunches collide head-on at the B0 and D0 interaction points but under go long range (electromagnetic) beam-beam interactions at 136 other locations around the ring[2] although any particular bunch only has long range interactions at 70 of these locations.

Running the simulation with all 136 long-range IPs turns out to be very slow so we only calculated beam-beam forces at the two main IPs and the long-range IPs immediately upstream and downstream of them. The transverse beta functions at the long-range collision locations are much larger than the bunch length, so the beam-beam calculation at those locations can be performed using only the 2D solver.

One interesting consequence of the fill pattern and the helical trajectory is that anyone of the 12 bunches in a train experiences collisions with the 36 bunches in the other beam at different locations around the ring, and in different transverse positions. This results in a different tune and emittance growth for each bunch of a train, but with the three-fold symmetry for the three trains. In the simulation, emittance growth arises from the effects of impedance acting on bunches that have been

[2]With three-fold symmetry of bunch trains, train-on-train collisions occur at six locations around the ring. The collision of two trains of 12 bunches each results in bunch-bunch collisions at 23 locations which, when multiplied by six, results in 138 collision points. It is a straightforward computer exercise to enumerate these locations. Two of these locations are distinguished as head-on while the remainder are parasitic. [17]

perturbed by beam-beam forces. The phenomenon of bunch dependent emittance growth is observed experimentally [17].

The beam-beam simulation with 36-on-36 bunches shows similar effects. We ran a simulation of 36 proton on 36 antiproton bunches for 50000 turns with the nominal helical orbit. The proton bunches had 8.8×10^{11} particles (roughly four times the usual number to enhance the effect) and the proton emittance was the typical 20π mm mrad. The antiproton bunch intensity and emittance were both half the corresponding proton bunch parameter. The initial emittance for each proton bunch was the same so changes during the simulation reflect the beam-beam effect.

Curve (a) in Figure 6 shows the emittance for each of the 36 proton bunches in a 36-on-36 simulation after 50000 turns of simulation. The three-fold symmetry is evident. The end bunches of the train (bunch 1, 13, 25) are clearly different from the interior bunches. For comparison, curve (c) shows the measured emittance taken during accelerator operations. The observed bunch emittance variation is similar to the simulation results. Another beam-beam simulation with the beam separation at the closest head-on IP expanded 100 times its nominal value resulted in curve (b) of Figure 6 showing a much reduced bunch-to-bunch variation. We conclude that the beam-beam effect at the long-range IPs is the origin of the bunch variation observed in the running machine and that our simulation of the beam helix is correct.

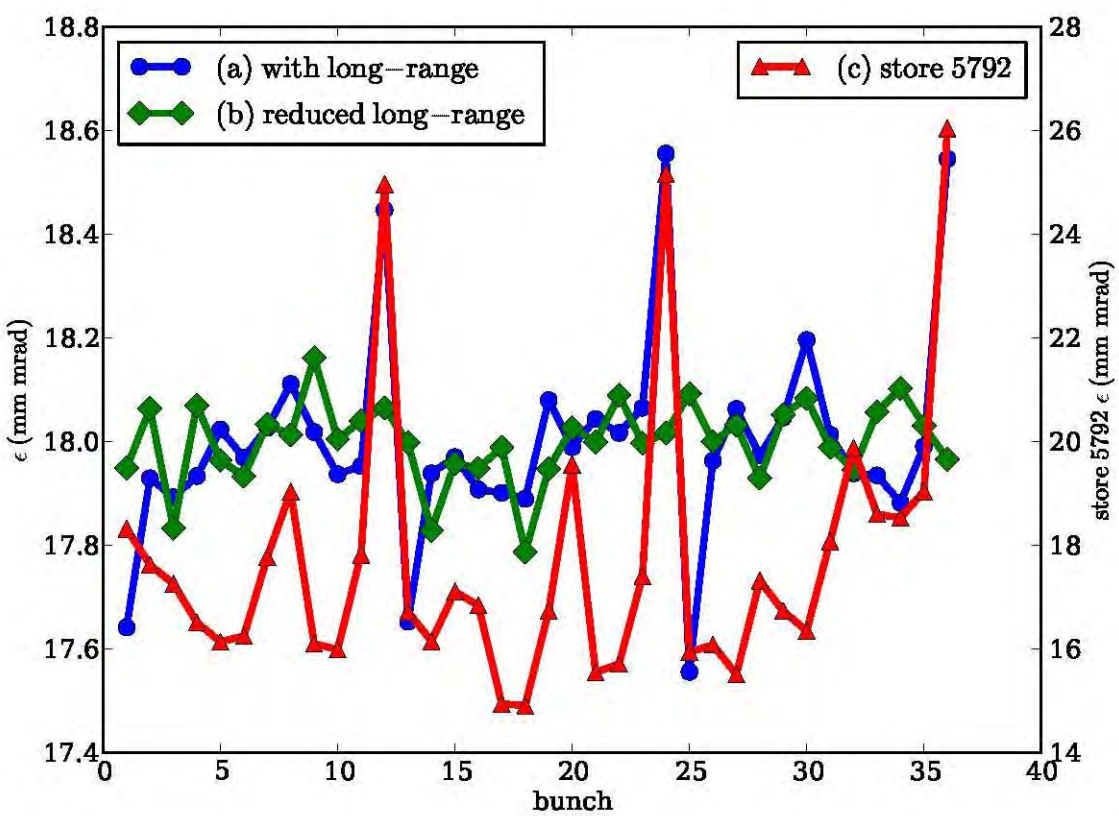


Figure 6. The simulated and measured emittance of each Tevatron proton bunch after running with 36 proton and 36 antiproton bunches. Curves (a) and (b) which show the emittance after 50000 simulated turns are read with the left vertical axis. Curve (a) results from a simulation with the nominal beam spacing at the long-range IPs. Curve (b) results from a simulation with the hypothetical condition where the beam separation at the long-range IPs is 100 times normal, suppressing the effect of those long-range IPs. Curve (c) is the measured emittance of bunches after 15 minutes of a particular store (#5792) of bunches in the Tevatron, and is read with the right vertical axis.

6. Lower Chromaticity Threshold

During the Tevatron operation in 2009 the limit for increasing the initial luminosity was determined by particle losses in the so-called squeeze phase [18]. At this stage the beams are separated in the main interaction points (not colliding head-on), and the machine optics is gradually changed to decrease the beta-function at these locations from 1.5 m to 0.28 m.

With proton bunch intensities currently approaching 3.2×10^{11} particles, the chromaticity of the Tevatron has to be managed carefully to avoid the development of a head-tail instability. It was determined experimentally that after the head-on collisions are initiated, the Landau damping introduced by beam-beam interaction would be strong enough to maintain beam stability at chromaticity of +2 units (in Tevatron operations, chromaticity is $\Delta\nu/(\Delta p/p)$.) At the earlier stages of the collider cycle, when beam-beam effects are limited to long-range interactions the chromaticity was kept as high as 15 units since the concern was that the Landau damping is insufficient to suppress the instability. At the same time, high chromaticity causes particle losses which are often large enough to quench the superconducting magnets hence it is desirable to keep chromaticity at a reasonable minimum.

Table 1. Beam parameters for Tevatron simulation

Parameter	value
beam energy	980 GeV
p particles/bunch	3.0×10^{11}
$\bar{p}$ particles/bunch	0.9×10^{11}
p tune (ν_x, ν_y)	(20.585,20.587)
p (normalized) emittance	20π mm mrad
$\bar{p}$ tune (ν_x, ν_y)	(20.577,20.570)
$\bar{p}$ (normalized) emittance	6π mm mrad
synchrotron tune ν_s	0.0007
slip factor	0.002483
bunch length (rms)	43 cm
$\delta p/p$ momentum spread	1.2×10^{-4}
effective pipe radius	3 cm

Table 2. The azimuthal location (s), beam offset (D_x, D_y), and beta function (β_x, β_y) of the important long-range collision locations simulated in the study of chromaticity reduction in the Tevatron. During the setup for collisions, the beam separation is several bunch widths at the $B0$ and $D0$ locations (see Fig. 2) where head-on collisions would occur during collider operations.

s (m)		D_x (μm)	D_y (μm)	β_x (m)	β_y (m)
-59.28		258	-661	164.49	16.46
0.00	$B0$	-167	-16	0.33	0.37
59.28		642	292	19.74	155.64
2035.12		263	721	182.10	15.65
2094.40	$D0$	4	-171	0.29	0.35
2153.67		-639	222	19.32	147.45

Our multi-physics simulation was used to determine the safe lower limit for chromaticity. The simulations were performed with starting beam parameters listed in Table 1. The major contributing long-range collision regions are listed in Table 2. Note that the beam separation is several bunch widths at the $B0$ and $D0$ region where collisions would occur during high-energy physics operations. With chromaticity set to –2 units, and no beam-beam effect, the beams are clearly unstable as seen in

Figure 7. With beams separated, turning on the beam-beam effect prevents rapid oscillation growth during the simulation as shown in Figure 8. The bursts of increased amplitude is sometimes indicative of the onset of instability, but it is not obvious within the limited duration of this run. The RMS size of the beam also does not exhibit any obvious unstable tendencies as shown in Figure 9.

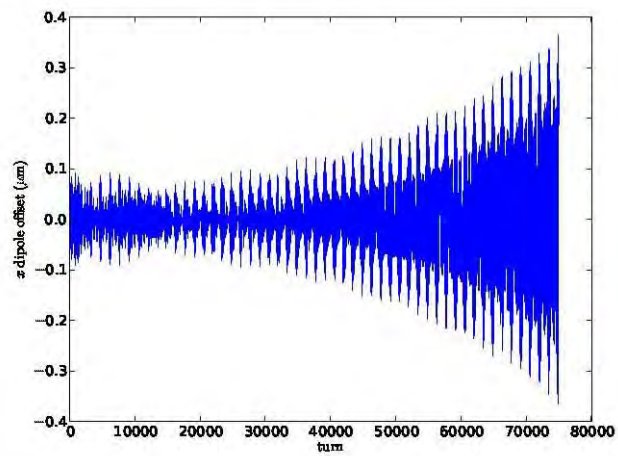


Figure 7. The x dipole moment in a simulation with $C = -2$ and no beam-beam effect showing the development of instability.

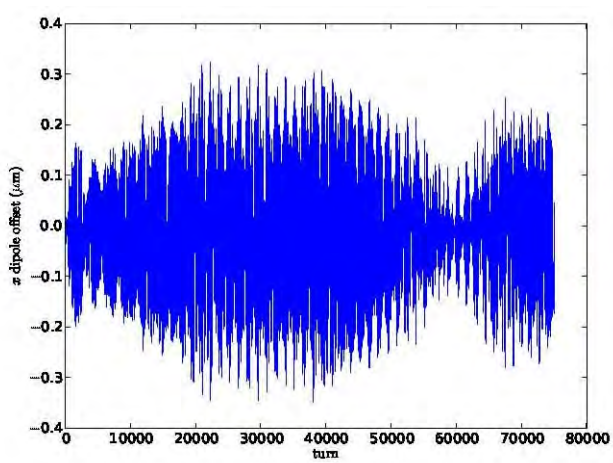


Figure 8. The x dipole moment of a representative bunch in a 36-on-36 simulation with $C = -2$ with beam-beam effects and beams separated showing no obvious instability within the limits of the simulation.

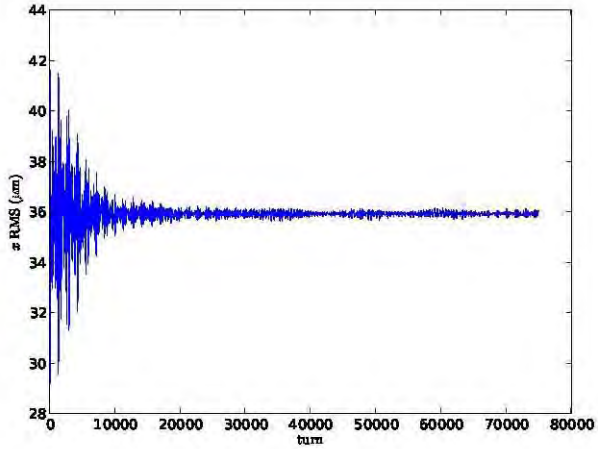


Figure 9. The x RMS moment of a representative bunch in a 36-on-36 simulation with $C = -2$ with beam-beam effects and beams separated showing no obvious instability within the limits of the simulation.

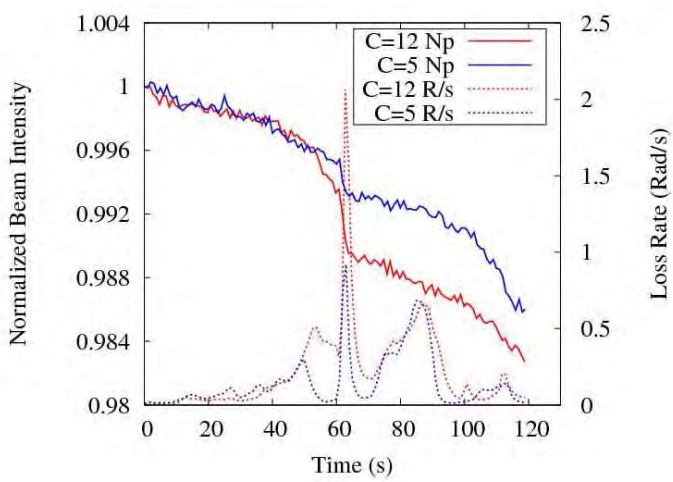


Figure 10. Plot of the beam current (solid line read with the left-hand scale) and radiation losses (dotted lines read with the right-hand scale) vs. time during the "squeeze" phase of setup for two values of chromaticity. The red curves are the values when running at a chromaticity $C = 12$ (the previously used high value) while the blue curves are for $C = 5$, the reduced chromaticity value. Lowering the chromaticity markedly prevents the drop in beam intensity around $t = 63$ and also lowers the radiation losses.

Based on these findings the chromaticity in the squeeze was lowered by a factor of two, and presently is kept at 8–9 units. This resulted in a 50% decrease in observed particle loss rates as shown in Figure 10 [18].

7. Summary

The key features of the simulation developed and improved under the SciDAC program include fully three-dimensional strong-strong multi-bunch beam-beam interactions with multiple interaction points, transverse resistive wall impedance, and chromaticity. The beam-beam interaction model has been shown to reproduce the location and evolution of synchrobetatron modes characteristic of the 3D strong-strong beam-beam interaction observed in experimental data from the VEPP-2M collider. The impedance calculation with macroparticles excites both the strong and weak head-tail instabilities with thresholds and growth rates that are consistent with expectations from a simple two-particle model and Vlasov calculation. Simulation of the interplay between the helical beam-orbit, long range beam-beam interactions and the collision pattern qualitatively matches observed patterns of emittance growth.

The new program is a valuable tool for evaluation of the interplay between the beam-beam effects and transverse collective instabilities. Simulations have been successfully used to support the change of chromaticity at the Tevatron, demonstrating that even the reduced beam-beam effect from long-range collisions may provide enough Landau damping to prevent the development of head-tail instability. These results were used in Tevatron operations to support a change of chromaticity during the transition to collider mode optics, leading to a factor of two decrease in proton losses, and thus improved reliability of collider operations.

Acknowledgments

We thank J. Qiang and R. Ryne of LBNL for the use of and assistance with the BeamBeam3d program. We are indebted to V. Lebedev and Yu. Alexahin for useful discussions. This work was supported by the United States Department of Energy under contract DE-AC02-07CH11359 and the ComPASS project funded through the Scientific Discovery through Advanced Computing program in the DOE Office of High Energy Physics. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

References

[1] US Particle Physics: Scientific Opportunities, Particle Physics Project Prioritization Panel, May 2008 available from http://www.er.doe.gov/hep/files/pdfs/P5_Report%2006022008.pdf.
[2] Run II handbook, http://www-bd.fnal.gov/runII.
[3] Evidence for an anomalous like-sign dimuon charge asymmetry, arXiv:1005.2757 [hep-ex] http://arxiv.org/abs/1005.2757, submitted to Phys. Rev. D.
[4] T. Aaltonen et al., (CDF and DCollaborations), Combined CDF and D Upper Limits on Standard Model Higgs Boson Production in the W+W-Decay Mode, Phys. Rev. Lett. 104, 061802 (2010)
[5] E. A. Perevedentsev and A. A. Valishev, "Simulation of Head-Tail Instability of Colliding Bunches", Phys. Rev. ST Accel. Beams 4, 024403, (2001)
[6] J. Qiang, M. A. Furman, R. D. Ryne, J. Comp. Phys., 198 (2004), pp. 278–294, J. Qiang, M. A. Furman, R. D. Ryne, Phys. Rev. ST Accel. Beams, 104402 (2002).
[7] V. Lebedev, http://www-bdnew.fnal.gov/pbar/organizationalchart/lebedev/OptiM/optim.htm
[8] A. Valishev et al., "Progress with Collision Optics of the Fermilab Tevatron Collider", EPAC06, Edinburgh, Scotland, 2006
[9] A. Chao, Physics of Collective Beam Instabilities in High Energy Accelerators., pp. 56–60, 178–187, 333-360 John Wiley and Sons, Inc., (1993)
[10] I. N. Nesterenko, E. A. Perevedentsev, A. A. Valishev, Phys. Rev. E, 65, 056502 (2002)
[11] K. Yokoya, Phys. Rev. ST-AB3, 124401 (2000).
[12] A. Piwinski, IEEE Trans. Nucl. Sci. NS-26 (3), 4268 (1979).

[13] E. A. Perevedentsev and A. A. Valishev, Phys. Rev. ST Accel. Beams 4, 024403 2001; in Proceedings of the 7th European Particle Accelerator Conference, Vienna, 2000 unpublished , p. 1223; http://accelconf.web.cern.ch/AccelConf/e00/ index.html
[14] V. V. Danilov and E. A. Perevedentsev, Nucl. Instrum. Methods Phys. Res. A 391, 77 1997.
[15] A. Chao, Physics of Collective Beam Instabilities in High Energy Accelerators., Figure 4.8 on p. 183, John Wiley and Sons, Inc., (1993).
[16] A. Chao, Physics of Collective Beam Instabilities in High Energy Accelerators., p. 351, John Wiley and Sons, Inc., (1993).
[17] V. Shiltsev, et al., "Beam-Beam Effects in the Tevatron," PRSTAB, 8, 101001 (2005).
[18] A. Valishev et al., "Recent Tevatron Operational Experience", PAC2009, Vancouver, BC, 2009

Computational and theoretical analysis of weak and strong transverse-wave structures in gaseous detonations

Ralf Deiterding[1]

[1]Oak Ridge National Laboratory, P.O. Box 2008 MS6367, Oak Ridge, TN 37831

E-mail: deiterdingr@ornl.gov

Abstract. Two-dimensional simulation results are presented that capture at great detail the temporal evolution of Mach reflection triple point sub-structures intrinsic to gaseous detonation waves. The observed patterns are classified by shock polar analysis for realistic, thermally perfect but nonreactive gases. A diagram of the transition boundaries between possible Mach reflection structures is constructed and found to be in good agreement with the computational results.

1. Introduction

The propagation of detonation waves in gaseous media is characterized by the appearance of transverse pressure waves propagating perpendicular to the detonation front. The hydrodynamic flow pattern around the resulting triple points is a Mach reflection phenomenon under transient conditions. Although the problem has received considerable attention (cf. [9]), no comprehensive theory is available yet to accurately predict the detailed multi-dimensional flow behavior.

In the present paper, we use high-resolution numerical simulation to analyze the hydrodynamic details of two-dimensional Chapman-Jouguet (CJ) detonations in low-pressure hydrogen-oxygen with high argon dilution. In free space, the triple point movement in such mixtures is very regular leading to a repetitive triple point trajectory pattern of regular “detonation cells” [8]. We study a series of computations in which the regularly oscillating detonation propagates through a smooth pipe bend of varying angle. It is found that nonreactive shock wave reflection theory [1], extended for thermally perfect gas mixtures, is applicable to classify the observed transient triple point structures as transitional and double Mach reflection patterns.

2. Computational Method

The used simulation code solves the Euler equations for multiple thermally perfect species with chemically reactive source terms. All species are assumed to be thermally perfect gases with temperature dependent enthalpy $h_i\,(T) = h_i^0 + \int_{To}^{T} c_{pi}\,(\sigma)\,d\sigma$ [6]. The chemical production rates are modeled with a hydrogen-oxygen reaction mechanism of 34 elementary reactions that considers the species H, O, OH, H_2, O_2, H_2O, HO_2, H_2O_2 and Ar [10].

A time-operator splitting approach is adopted to decouple hydrodynamic transport and chemical reaction numerically. A semi-implicit Rosenbrock-Wanner method is used to integrate the kinetics, and a shock-capturing finite volume discretization is employed to achieve proper upwinding in all characteristic fields. The finite volume scheme utilizes a quasi-one-dimensional approximate Riemann solver of Roe-type and is extended to multiple space dimensions via the method of fractional steps.

Special corrections are applied to avoid unphysical total densities and internal energies, to ensure positive mass fractions, and to prevent the carbuncle phenomenon. The MUSCL-Hancock variable extrapolation technique is adopted to construct a second-order method (see [5] for details).

Geometrically complex moving boundaries are incorporated into the originally Cartesian upwind method by utilizing some of the finite volume cells as ghost cells to enforce immersed rigid wall boundary conditions [5]. Slight approximation errors due to this approach are alleviated by dynamic mesh adaptation.

Since the accurate numerical simulation of detonation waves mandates a particularly high temporal and spatial resolution near the detonation front, we utilize the structured adaptive mesh refinement (SAMR) algorithm after Berger and Colella [2]. We have implemented the SAMR algorithm in a generic, dimension-independent object-oriented framework in C++ called AMROC. AMROC follows a rigorous domain decomposition approach, suitable for MPI-based distributed memory parallelization, and partitions the SAMR hierarchy from the root level on [3]. The software system has been verified extensively for shock and detonation wave simulation [4]. The latest version exhibits good parallel scalability on $O(1000)$ cores.

3. Results

We study two-dimensional, self-sustained CJ detonations in a H_2 : O_2 : Ar mixture of molar ratios 2 : 1 : 7 at initial temperature 298K and pressure 10.0 kPa. The one-dimensional detonation theory after Zel'dovich, von Neumann, and Döring (ZND) predicts a detonation velocity of 1638.5 m/s, a von Neumann Pressure of ~270 kPa, and a reaction length of 0.878 mm for this mixture. All shown computations use four additional levels of Cartesian mesh adaptation with refinement factors 2, 2, 2, and 4 giving an effective resolution of 67.6 cells within the half reaction length, the distance between shock and reaction zone according to ZND theory. Mesh adaptation is based on a physically motivated combination of scaled gradients of total density and pressure and error estimation by Richardson extrapolation of the mass fractions (cf. [5]).

3.1. Regular Cellular Detonation Structure

The results shown in Figure 1 use as computational domain a square channel of the physical dimensions 1.0 m × 3.2 cm, in which four transverse pressure waves and thereby two perfectly regular detonation cells occur. The base mesh has 2000 × 128 cells. The left graphic of Figure 1 displays triple point trajectories in a part of the computational domain with the detonation front at two time steps overlaid. The trajectories are visualized by tracking the maximum of the magnitude of the vorticity on a uniform auxiliary grid with the mesh widths of the first refinement level. The right graphic of Figure 1 shows the different level of adaptation, indicated by gray scales, for the first time step.

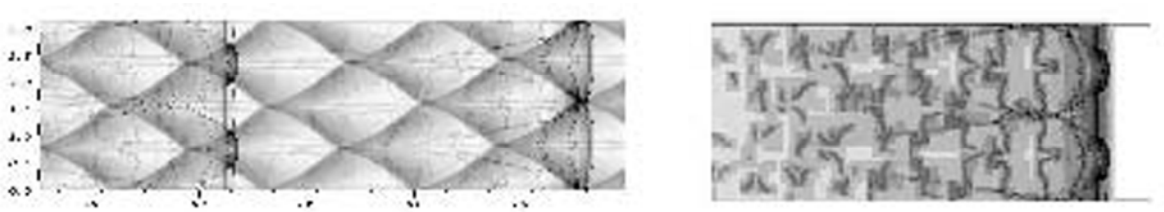
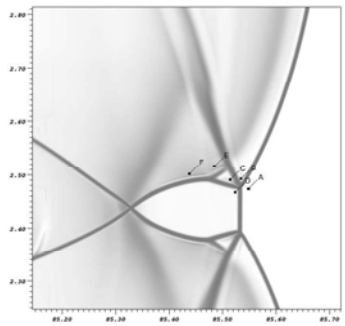


Figure 1. Left: oscillating detonation front on computed triple points tracks in a rectangular channel. Right: schlieren plot of density on refinement levels.

Figure 2 zooms into the situation shortly before the collision of two triple points. The middle enlargement shows clearly that the shock wave pattern around each triple point is of double Mach reflection (DMR) (aka "strong") type. As depicted in the right sketch, the essential flow features around triple point T are: inflow (A), region B behind the Mach stem M, transverse wave region (C),

	p/p_A	ρ/ρ_A	T [K]	v[m/s]	M
A	1.00	1.00	298	1775	5.078
B	31.45	4.17	2248	447	0.477
C	31.69	5.32	1775	965	1.153
D	19.17	3.84	1487	1178	1.533
E	35.61	5.72	1856	901	1.053
F	40.61	6.09	1987	777	0.880



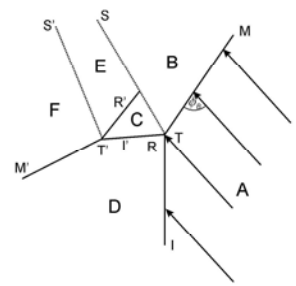


Figure 2. States of a DMR structure shortly before the next collision.

and region D downstream of the incident shock I. Regions B and C are separated by the slip line S, a contact discontinuity. Regions C and D are separated by the reflected shock R. Characteristic for the DMR pattern is a high supersonic velocity in region C that leads to the formation of a further shock R' creating a secondary triple point T' on the transverse wave. The very weak slip line S', emanating from T' between regions E and F, can hardly be inferred. Note that the reflected shock R is the incident shock I' for the secondary triple point T'.

3.2. Transformation into the Frame of Reference of a Triple Point

In order to analyze a Mach reflection pattern quantitatively, it is necessary to map the velocity field of the simulation into a frame of reference attached to the triple point. However, the reliable estimation of the triple point speed v_0 from a single computational time step is non-apparent. When evaluating v_0 we take advantage of the fact that the triple point is formed at the tip, where the Mach stem intersects the incident region, and that the oblique shock relations [1] between two points in regions A and B close to the triple point must hold true. We require only the two relations

$$\rho_A v_A \sin(\phi_B) = \rho_B v_B \sin(\phi_B - \theta_B) , \quad (1)$$

$$p_A + \rho_A v_A^2 \sin^2(\phi_B) = p_B + \rho_B v_B^2 \sin^2(\phi_B - \theta_B) . \quad (2)$$

Inserting Eq. (1) into Eq. (2) allows the elimination of $v_B \sin(\phi_B - \theta_B)$, which yields

$$v_A = \frac{1}{\sin\phi_B}\sqrt{\frac{\rho_B(p_B - p_A)}{\rho_A(\rho_B - \rho_A)}} .$$

As the gas is initially at rest, the triple point velocity is $v_0 = -v_a$ and ϕ_B, the angle of inflow, is given as the angle between Mach stem front and the triple point trajectory, which can be measured from visualizations such as the left one of Figure 1. The states close of the triple point of Figure 2 are given in the table of Figure 2. Since the triple point is far ahead of the reaction region, changes in mixture are neglected in evaluating the Mach number M in the triple point pattern. As it can be expected in a DMR [1], M_C is clearly greater than 1.

A further important quantity for triple point structures is the strength S of the transverse wave [7] that is defined as $S := \frac{pC-pD}{pD}$. For the present computation, S is rather constant throughout one regular detonation cell and varies around 0.65.

3.3. Detonation Structure in Smooth Pipe Bends

In order to study triple point structures under transient conditions, we simulate the propagation of the regularly oscillating detonation through smooth pipe bends of varying angles. The computations are initialized by reproducing a single detonation cell of the previous computation five times 13 cm before the beginning of the curved section. For instance, for bend angle $\varphi = 60°$, with a base mesh of 1200 ×

992 cells, the adaptive computation uses approximately 7.1 M to 3.4 M cells on all and 4.8 M to 1.8 M cells on the highest level instead of ~1, 219 M in the uniform case. The calculations were run on 64 Intel Xeon 2.4GHz dual-processor nodes with Quadrics interconnect and required nevertheless ~70,000 h CPU each (~23 days wall time).

The diffraction and compression of the detonation wave at the bend leads to changes of the transverse wave strength *S*. While the state *C* behind the reflected shock *R* remains initially unchanged, the change in geometry alters the incident state *D*. Near the inner bend wall, the incident pressure p_D drops, leading to a considerable increase in *S*. The resulting triple point structure is of strengthened DMR type with larger detonation cells (cf. Right graphic of Figure 3) and increased M_C. In the compression region near the outer wall, however, p_D increases drastically leading to a considerable decrease of *S*. As a consequence, the flow in region *C* decelerates and the secondary shock *R'* between *C* and *E* is no longer necessary for a stable configuration. The triple point now exhibits a transitional Mach reflection (TMR) or "weak" pattern. Characteristic for the TMR structure is that the flow in region *C* is then just barely supersonic with respect to *T* [1]. TMR structures are also found in the diffraction region when a sufficiently large angle φ causes partial detonation failure.

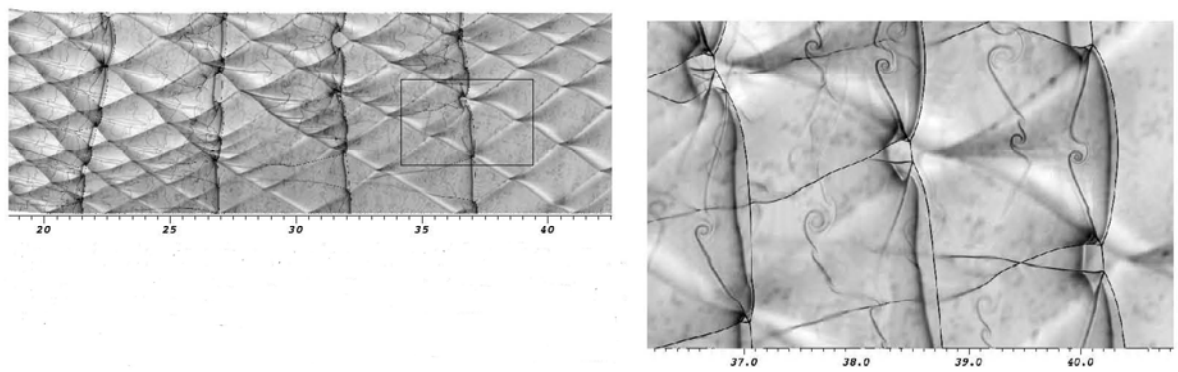


Figure 3. Left: cellular structure after the bend for φ = 15°. Right: triple point re-initiation with change from TMR to DMR. The physical time steps 200 μs, 210 μs, and 220 μs are shown.

The right graphic of Figure 3 depicts the interesting case of triple point re-initiation after the bend for φ = 15°. The new triple point in the center of the graphic is formed as a weak structure (TMR) but strengthens into a DMR after its first collision. Figure 4 visualizes the quenching of triple points at the outer bend wall that occurs for all larger values of φ. As can be inferred from the left graphic of Figure 4, failing triple points with vanishing trajectories seem to exhibit a weak structure.

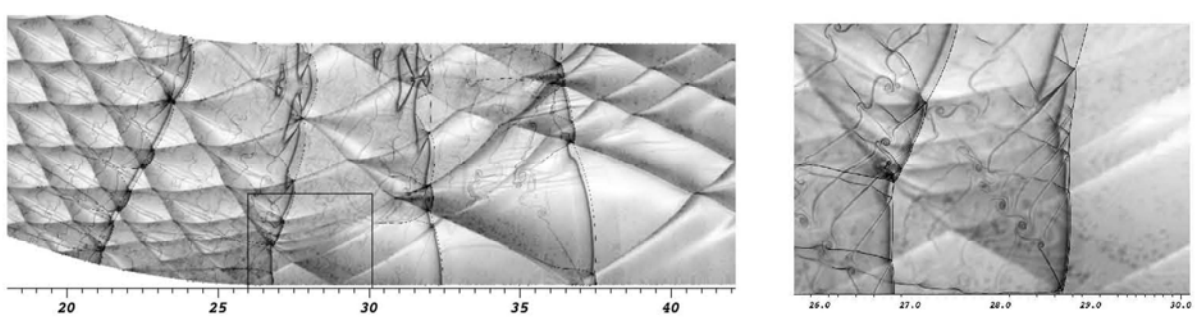


Figure 4. Left: cellular structure immediately after the bend for φ = 30°. Right: triple point quenching and failure as SMR at 140 μs and 150 μs simulated physical time.

3.4. Transition Criteria of Mach Reflection Phenomena

The two dimensional oblique jump conditions [1] are satisfied across each discontinuity depicted in the right sketch of Figure 2. We solve the resulting nonlinear system of equations numerically under the assumption that the composition of the H_2 : O_2 : Ar mixture is invariant, but note that the temperature dependency of the material properties is fully considered. It is well-known (cf. [1]) that changes in Mach reflection type occur when the flow in certain regions of the triple point structure becomes supersonic in the frame of reference of the triple point T or T'. Any Mach reflection requires $M_B^T > 1$; for $M_B^T < 1$ a regular reflection (RR) occurs. A single Mach reflection (SMR) pattern encompassing just the four states A to D is found for $M_C^T < 1$; for $M_C^T > 1$ a transitional or a double Mach reflection pattern occurs. In a DMR, the flow in C is also supersonic in the frame of reference of the secondary triple point $T'(M_C^{T'} > 1)$, however, it is subsonic with $M_C^{T'} < 1$, in a TMR. Consequently, the secondary triple point T' is not fully developed in a TMR and the discontinuities R' and S' are absent. For given inflow velocity v_0 the transition points $M_{B/C}^T = 1$ are unique and can be determined rather easily by solving the corresponding set of oblique shock relations. The evaluation of the TMR/DMR transition point $M_C^{T'} = 1$ is more involved as knowledge of the velocity vector a of T' relative to T is required. The oblique shock relations between C and D readily yield $a_n \equiv 0$, however, the tangential velocity a_t of T' along R is a free parameter that needs to be specified separately.

Figure 5 visualizes the transition boundaries of the different shock wave reflection phenomena evaluated numerically for varying inflow Mach number. Since the estimation of the exact minimally possible value of a_t in a DMR is a considerable problem in itself, the TMR/DMR transition line is evaluated for the fixed value a_t = 100 m/s, which puts a reasonable upper transition boundary on the TMR region (dotted in Figure 5). Specially geared to detonations, results are displayed for the relative transverse wave strength S. We use the simulation results from Sections 3.1 and 3.3 as a first verification for the found quantitative transition boundaries. Throughout an unperturbed detonation cell only the DMR structure occurs, which is confirmed by plotting (with stars) beside the point at the end of a cell (cf. Figure 2) an additional one immediately after triple point collision. The open circles mark the triple point re-initiation visualized in the right graphic of Figure 3, and the transition boundaries clearly confirm the transition from TMR to DMR. The failing triple point shown in the right graphic of Figure 4 is marked with closed triangles indicating that the SMR pattern seems unstable for a detonation substructure and occurs only immediately before the triple point disappears.

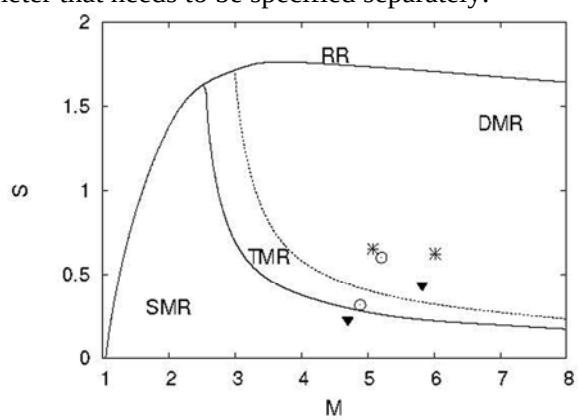


Figure 5. Transition boundaries of shock wave reflection phenomena for the nonreactive H2 : O2 : Ar mixture.

4. Conclusions

We have presented simulation results for gaseous detonations in two space dimensions in realistic geometry. The computations are fully resolved down to the scale of secondary triple points and can be used with confidence to analyze the evolving shock wave reflection structures. A crucial enabler for these simulations is the utilization of MPI-parallel dynamic structured adaptive mesh refinement which reduces the finest mesh size for the pipe bend studies by at least a factor of 250 and up to 680.

Nonreactive but thermally perfect shock wave reflection theory is used successfully to predict the transition boundaries between Mach reflection patterns depending on the inflow velocity. Exemplary transient triple points structures from the simulations are compared with the theoretical results and

good agreement in observed and predicted Mach reflection type is found. Future work will concentrate on deriving a physical estimate for the smallest possible relative tangential velocity a_t between primary and secondary triple point in a double Mach reflection to complete the theoretical model.

Acknowledgments
This work is sponsored by the Office of Advanced Scientific Computing Research; U.S. Department of Energy (DOE) and was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725.

References

[1] G. Ben-Dor. *Shock wave reflection phenomena*. Springer, Berlin, 2nd edition, 2007.
[2] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1988.
[3] R. Deiterding. Construction and application of an AMR algorithm for distributed memory computers. In T. Plewa et, editors, *Adaptive Mesh Refinement—Theory and Applications*, volume 41 of *Lecture Notes in Computational Science and Engineering*, pages 361–372. Springer, 2005.
[4] R. Deiterding. Detonation structure simulation with AMROC. In L. T. Yang, editor, *High Performance Computing and Communications 2005*, volume 3726 of *Lecture Notes in Computer Science*, pages 916–927. Springer, 2005.
[5] R. Deiterding. A parallel adaptive method for simulating shock-induced combustion with detailed chemical kinetics in complex domains. *Computers & Structures*, 87:769–783, 2009.
[6] R. Deiterding and G. Bader. High-resolution simulation of detonations with detailed chemistry. In G. Warnecke, editor, *Analysis and Numerics for Conservation Laws*, pages 69–91. Springer, 2005.
[7] W. Fickett and W. C. Davis. *Detonation*. University of California Press, Berkeley and Los Angeles, California, 1979.
[8] X.Y. Hu, B.C. Khoo, D.L. Zhang, and Z.L. Jiang. The cellular structure of a two dimensional $H_2/O_2/Ar$ detonation wave. *Combustion Theory & Modelling*, 8:339–359, 2004.
[9] R.A. Strehlow. Gas phase detonations: Recent developments. *Combust. Flame*, 12(2):81–101, 1968.
[10] C.K. Westbrook. Chemical kinetics of hydrocarbon oxidation in gaseous detonations. *Combust. Flame*, 46:191–210, 1982.

Numerical Simulation of Phase Space Advection in Gyrokinetic Models of Fusion Plasmas

M. R. Dorr[1], R. H. Cohen[2], P. Colella[3], M. A. Dorf[2], J. A. F. Hittinger[1], and D. F. Martin[3]

[1]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 7000 East Avenue L-561, Livermore, CA 94550[*]
[2]Fusion Energy Program, Lawrence Livermore National Laboratory, 7000 East Avenue L-630, Livermore, CA 94550. [3]Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, One Cyclotron Road Mail Stop 50A-1148, Berkeley, CA 94720[†]

Abstract. We describe our progress in the development of a fourth-order, finite-volume discretization of a nonlinear, full-f gyrokinetic Vlasov-Poisson system in mapped coordinates. The approach treats the configuration and velocity components of phase space on an equal footing, using a semi-discretization with limited centered fluxes combined with a fourth-order Runge Kutta integration. The algorithm has been implemented in a new code named COGENT, which is built on the Chombo Adaptive Mesh Refinement library being developed by the SciDAC APDEC project. The algorithm and code have been tested on geodesic acoustic mode problems in a standard equilibrium magnetic geometry by comparing damping rates and frequencies with those predicted by theoretical dispersion analyses. Convergence tests are also performed to verify the expected increase in accuracy as the phase space grid was refined.

1. Introduction

The ability to computationally model the behavior of the edge plasma in fusion reactors is a key component in the development of a whole device predictive simulation capability.

Among the features that distinguish the edge from the core is the development of a region of steep gradients in the density and temperature profiles called the pedestal, the height of which determines the quality of plasma confinement, and hence fusion gain. A kinetic plasma model in needed in this region (Figure 1), because the radial width of the pedestal observed in experiments is comparable to the radial width of individual particle orbits (leading to large distortions of the local distribution functions from a Maxwellian), while the mean free path can be comparable to the scale length for temperature variations along the magnetic field (violating the assumptions underlying a collisional fluid model).

Because of the large number of independent variables in a fully kinetic model, as well as the fast time scale represented by the ion gyrofrequency, gyrokinetic models (which average over fast gyro motion about field lines) have been developed to facilitate numerical treatments. Continuum models consist of a Boltzmann equation for evolving plasma species distribution functions in a particular coordinate system

[*] This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344
[†] Research supported by the Office of Advanced Scientific Computing Research of the US Department of Energy under contract number DE-AC02-05CH11231.

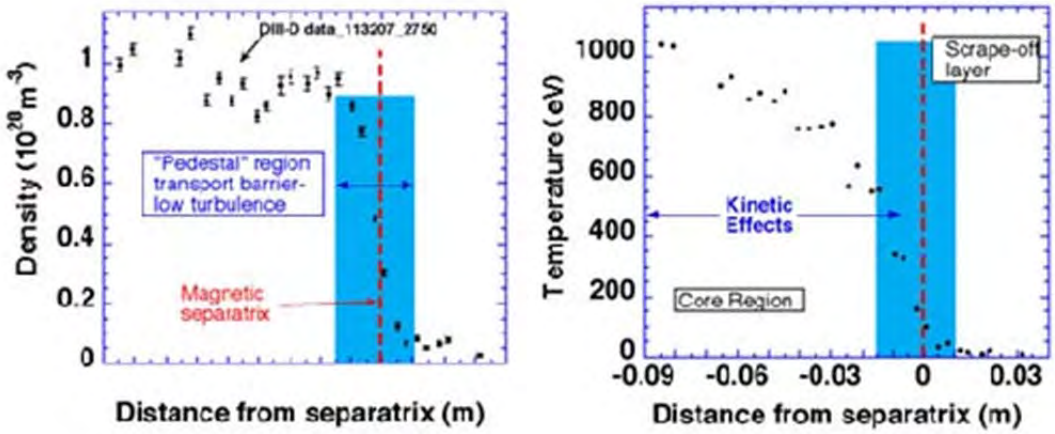


Figure 1. DIII-D pedestal [13].

combined with some variant of Maxwell's equations. Gyrokinetic codes such as GENE[9, 10], GS2[6,11] and GYRO[1, 3, 2] have been successfully employed to model core plasmas for many years. In addition to requiring simpler geometries, these codes exploit the fact that in the core, distribution functions are typically small perturbations δf about a known Maxwellian distribution f_0, providing a simpler, and even sometimes linear, model. To model the edge plasma all the way to the reactor walls, a method to solve nonlinear gyrokinetic models for the entire distribution function (so-called full-f) in edge-relevant geometries is needed.

In this paper, we summarize our progress to date in the development and application of new algorithmic methodologies in the solution of a nonlinear, full-f, gyrokinetic Vlasov-Poisson system. Beginning with a conservative formulation, the system is treated as a nonlinear advection equation in a 4D (2 configuration space + 2 velocity space coordinates) or 5D (3 configuration space + 2 velocity space coordinates) phase space. A semi-discretization is employed based on a recently developed formalism [4] for the creation of arbitrarily high-order finite-volume spatial discretizations in mapped coordinates. In the present context, mapped coordinates enable the use of a multiblock, locally rectangular computational domain, one of whose coordinates is aligned with magnetic flux surfaces. The use of a finite volume formulation naturally enables the discrete enforcement of conservation, whereas a fourth-order discretization provides for more efficient gridding of phase space and reduced numerical dissipation for long-time integration. We demonstrate the performance of the algorithms on the simulation of geodesic acoustic modes, which are eigenmodes of the gyrokinetic Vlasov-Poisson system.

2. The gyrokinetic Vlasov-Poisson system

We adopt the full f gyrokinetic model of [8]:

$$\frac{\partial(B_{\parallel\alpha}^{*} f_\alpha)}{\partial t} + \boldsymbol{\nabla}_{\mathbf{R}} \cdot \left(\dot{\mathbf{R}}_\alpha B_{\parallel\alpha}^{*} f_\alpha\right) + \frac{\partial}{\partial v_\parallel}\left(\dot{v}_{\parallel\alpha} B_{\parallel\alpha}^{*} f_\alpha\right) = 0, \tag{1}$$

where

$$\dot{\mathbf{R}}_\alpha \equiv \dot{\mathbf{R}}_\alpha(\mathbf{R}, v_\parallel, \mu, t) \equiv \frac{v_\parallel}{B_{\parallel\alpha}^{*}}\mathbf{B}_\alpha^{*} + \frac{\rho_L}{Z_\alpha B_{\parallel\alpha}^{*}}\mathbf{b} \times \mathbf{G}_\alpha, \tag{2}$$

$$\dot{v}_{\parallel\alpha} \equiv \dot{v}_{\parallel\alpha}(\mathbf{R}, v_\parallel, \mu, t) \equiv -\frac{1}{m_\alpha B_{\parallel\alpha}^{*}}\mathbf{B}_\alpha^{*} \cdot \mathbf{G}_\alpha, \tag{3}$$

and

$$\mathbf{B}_\alpha^* \equiv \mathbf{B}_\alpha^*(\mathbf{R}, v_\parallel) \equiv \mathbf{B} + \rho_L \frac{m_\alpha v_\parallel}{Z_\alpha} \nabla_\mathbf{R} \times \mathbf{b}, \tag{4}$$

$$B_{\parallel_\alpha}^* \equiv B_{\parallel_\alpha}^*(\mathbf{R}, v_\parallel) \equiv \mathbf{b} \cdot \mathbf{B}_\alpha^*, \tag{5}$$

$$\mathbf{G}_\alpha \equiv \mathbf{G}_\alpha(\mathbf{R}, \mu, t) \equiv Z_\alpha \nabla_\mathbf{R} \Phi + \frac{\mu}{2} \nabla_\mathbf{R} B. \tag{6}$$

The unknown quantity $f_\alpha \equiv f_\alpha(\mathbf{R}_\alpha, v_{||}, \mu, t)$ is the distribution function of the αth plasma species in gyrocenter phase space coordinates ($\mathbf{R}_\alpha$, $v_{||}$, μ), whose equations of motion are given by (2)-(6). To simplify the exposition here, we have neglected the drift due to the equilibrium potential. We have also assumed a particular normalization that nondimensionalizes all quantities. For example, the quantity ρ_L appearing in the last term of (2) denotes the ion Larmor radius of the normalizing reference species relative to the chosen normalizing length scale.

Gyrocenter coordinates play a key role in gyrokinetic models in two important ways. First, they reduce what would otherwise be a six-dimensional phase space to five dimensions: $\mathbf{R}_\alpha$ is the three-dimensional configuration space coordinate, $v_{||}$ is the velocity space component along field lines, and the magnetic moment $\mu \equiv m_\alpha v_\perp^2 / 2B$ is related to the velocity $v_\perp$ perpendicular to field lines. Through the use of asymptotic orderings, gyrocenter coordinates are specifically constructed so as to make the distribution function f symmetric with respect to gyrophase. The latter component, which would have been the third velocity component, can then be ignored. The magnetic moment μ, an adiabatic invariant, is assumed to be constant in the development of gyrokinetic theories, which is why no evolution equation appears for it. The second benefit of gyrocenter coordinates is that the gyrofrequency is eliminated, which would otherwise represent a fast time scale that would need to be resolved. Because gyrocenter coordinates are developed as a Hamiltonian dynamical system, they satisfy the area preserving property

$$\nabla_\mathbf{R} \cdot \left(B_{\parallel_\alpha}^* \dot{\mathbf{R}}_\alpha \right) + \frac{\partial}{\partial v_\parallel} \left(B_{\parallel_\alpha}^* \dot{v}_{\parallel_\alpha} \right) = 0, \tag{7}$$

where $B_{||}^*_\alpha$ is the Jacobian of the mapping between lab frame and gyrocenter coordinates. As noted in [8], the gyrokinetic Vlasov equation can therefore be expressed in either convective or conservative form. We choose the latter with the objective of achieving a correspondingly conservative numerical discretization.

The potential Φ in (6) is evaluated by solving some form of Maxwell's equations. Here, we assume an electrostatic model obtained by the addition of Poisson's equation to the Vlasov system (1)-(6). The potential then depends only upon the charge density of the distribution functions f_α. A difficulty arises, however, from the fact that the f_α are only known in gyrocenter coordinates, whereas the Poisson equation is posed in the lab frame. The velocity integral used to compute the ion charge density must therefore be split into two pieces. In the long wavelength limit $k\rho \ll 1$, where k is the magnetic field wave number and ρ is the ion gyroradius, the gyrokinetic Poisson equation is

$$\nabla_\mathbf{X} \cdot \left\{ \left[\lambda_D^2 \mathbf{I} + \rho_L^2 \sum_\alpha \frac{Z_\alpha \bar{n}_\alpha}{m_\alpha \Omega_\alpha^2} \left(\mathbf{I} - \mathbf{b}\mathbf{b}^T \right) \right] \nabla_\mathbf{X} \Phi \right\} = n_e - \sum_\alpha Z_\alpha \bar{n}_\alpha, \tag{8}$$

where $\nabla_\mathbf{X}$ denotes the gradient with respect to the normalized lab frame coordinate and λ_D is the normalized Debye length.

$$\bar{n}_\alpha(x,t) \equiv \frac{1}{m_\alpha} \int f_\alpha(x, v_\parallel, \mu, t) B_\parallel^*(x, v_\parallel) dv_\parallel d\mu \tag{9}$$

is the ion gyrocenter density, which is the gyrophase independent part of the integration of the gyrocenter distribution function f_α over velocity. The second term in the left-hand side of (8) is the polarization density, which is the gyrophase-dependent part of the velocity integration of f_α . Since this piece depends upon the potential, we must combine it with the usual Laplacian (the first term in (8)) in the construction of the linear operator to be solved for Φ. Here, **b** denotes the unit vector in the direction of the magnetic field, Z_α is the charge state, m_α is the mass and Ω_α is the gyrofrequency. We note that for typical tokamak parameters, $\lambda_D \ll \rho_L$, and hence the polarization density term dominates. Because the electron gyroradius is small, a similar splitting of the electron density is omitted.

The gyrokinetic Vlasov-Poisson system is posed in a domain defined by the tokamak magnetic geometry, which is comprised of field lines lying on concentric flux surfaces. Since there are large variations of plasma parameters along and across field lines, there is strong motivation to discretize in coordinates where at least one of the coordinate directions is defined by the flux surfaces. As depicted in Figure 2, a natural choice is a mapped multiblock coordinate system, where the blocks correspond to the logically distinct core, scrapeoff layer and private flux regions. Within each block, a rectangular coordinate system can be employed, which facilitates efficient and accurate discretizations and domain decompositions over processors.

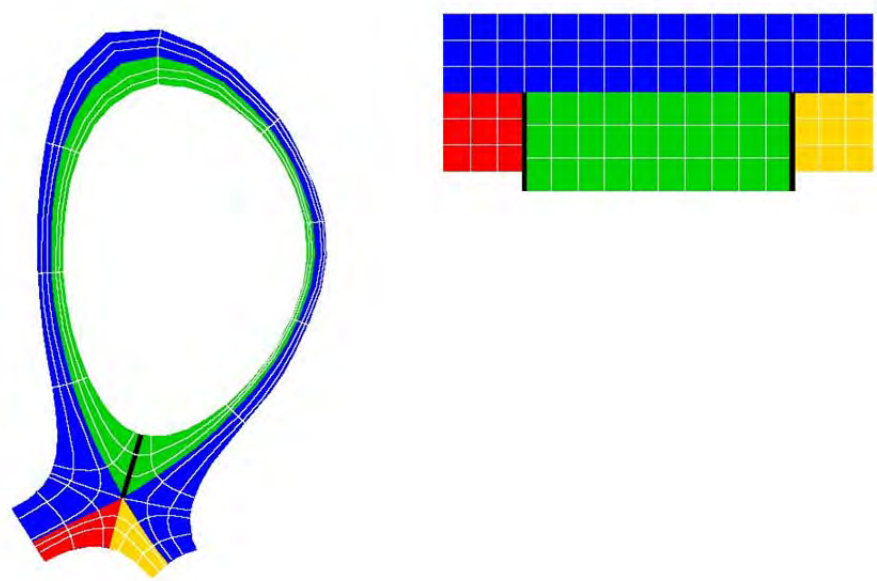


Figure 2. Single null geometry (left) and multi-block, locally rectangular computational domain (right).

We are therefore led to consider the discretization of the gyrokinetic Vlasov-Poisson system (1)-(6) in a mapped coordinate system. Among our requirements is a conservative formulation, thereby discretely expressing the phase space conservation law (1). A second requirement is high-order accuracy, enabling a more efficient deployment of degrees of freedom in discretizing the high-dimensional phase space. A high-order method is also important for reducing numerical dissipation in long-time integrations. Additional requirements include enforcement of positivity for the distribution functions f_α and an efficient solver for the gyrokinetic Poisson equation.

3. High-order, finite volume discretization in mapped coordinates

Next, we summarize a general approach for the systematic development of high-order finite volume discretizations in mapped coordinates. More complete details are contained in [4].

Suppose that we have a smooth mapping $\boldsymbol{X}$ from the unit cube onto the spatial domain Ω:

$$\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{\xi}),\ \boldsymbol{X} : [0,1]^3 \to \Omega.$$

Given this mapping, the divergence of a vector field on Ω can be written in terms of derivatives in $[0, 1]^3$, which will serve as our computational domain. That is,

$$\boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{F} = \frac{1}{J} \boldsymbol{\nabla}_{\boldsymbol{\xi}} \cdot (\boldsymbol{N}^T \boldsymbol{F}), \tag{10}$$

$$J = det(\boldsymbol{\nabla}_{\boldsymbol{\xi}} \boldsymbol{X})\ ,\ \boldsymbol{N}^T_{p,q} = det((\boldsymbol{\nabla}_{\boldsymbol{\xi}} \boldsymbol{X})(p|\boldsymbol{e}^q)), \tag{11}$$

where $\boldsymbol{A}(p|v)$ denotes the matrix obtained by replacing the pth row of the matrix $\boldsymbol{A}$ by the vector v and e^q is the unit vector in the q-th direction. The relationship (10) is an easy consequence of the chain rule, equality of mixed partials, and Cramer's rule.

In a finite volume approach, Ω is discretized as a union of control volumes. For Cartesian grid finite volume methods, a control volume Vi takes the form

$$V_{\boldsymbol{i}} = \left[\left(\boldsymbol{i} - \frac{1}{2}\boldsymbol{u} \right) h, \left(\boldsymbol{i} + \frac{1}{2}\boldsymbol{u} \right) h \right]\ ,\ \boldsymbol{i} \in \mathbb{Z}^3\ ,\ \boldsymbol{u} = (1,1,1),$$

where h is the grid spacing. When using mapped coordinates, we define control volumes in Ω as the images $X(V_i)$ of the cubic control volumes $V_i \subset [0, 1]^3$. Then, by changing variables and applying the divergence theorem, we obtain the flux divergence integral over a physical control volume $X(V_i)$ by

$$\int_{\boldsymbol{X}(V_{\boldsymbol{i}})} \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{F} d\boldsymbol{x} = \int_{V_{\boldsymbol{i}}} \boldsymbol{\nabla}_{\boldsymbol{\xi}} \cdot (\boldsymbol{N}^T \boldsymbol{F}) d\boldsymbol{\xi} = \sum_{\pm=+,-} \sum_{d=1}^{3} \pm \int_{A_d^{\pm}} (\boldsymbol{N}^T \boldsymbol{F})_d dA_{\boldsymbol{\xi}}, \tag{12}$$

where the $A_d^{\pm}$ are upper and lower faces of cell V_i in the d-th direction. As described in [4], the integrals on the cell faces $A_d^{\pm}$ can be approximated using the following formula for the average of a product in terms of fourth-order accurate face averages of each factor:

$$\langle fg \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle f \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \langle g \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{h^2}{12} \boldsymbol{G}_0^{\perp,d} \left(\langle f \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \right) \cdot \boldsymbol{G}_0^{\perp,d} \left(\langle g \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \right) + O(h^4). \tag{13}$$

Here, $G_0^{\pm,d}$ is the second-order accurate central difference approximation to the component of the gradient operator orthogonal to the d-th direction: $G_0^{\pm,d} \approx \nabla\xi - e^d \frac{\partial}{\partial \xi d}$, and the operator $<\cdot> i + \frac{1}{2}e^d$ $<\cdot> i + \frac{1}{2}e^d$ denotes a fourth-order accurate average over the face centered at $i + \frac{1}{2}e^d$:

$$\langle q \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{h^2} \int_{A_d} q(\boldsymbol{\xi}) dA_{\boldsymbol{\xi}} + O(h^4).$$

Alternative expressions to (13) are obtained by replacing the averages $< \int > i + \frac{1}{2}e^d$ and/or $< g > i + \frac{1}{2}e^d$ used in the transverse gradients $G_0^{\perp,d}$ by the corresponding face-centered point-wise values $< \int > i + \frac{1}{2}e^d$ and/or $g\ i + \frac{1}{2}e^d$, respectively.

We therefore have

$$\int_{\boldsymbol{X}(V_i)} \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{F} d\boldsymbol{x} = h^2 \sum_{d=1}^{3} \sum_{\pm=+,-} \pm F^d_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} + O(h^4), \tag{14}$$

where

$$F^d_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} \equiv \sum_{s=1}^{3} \langle N^s_d \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \langle F^s \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{h^2}{12} \sum_{s=1}^{3} \left(\boldsymbol{G}_0^{\perp,d} \langle N^s_d \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \right) \cdot \left(\boldsymbol{G}_0^{\perp,d} (\langle F^s \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}) \right), \tag{15}$$

F^s is the s-th component of $\boldsymbol{F}$ and N^s_d is the (s, d)-th element of the matrix $\boldsymbol{N}$. In [4], it is demonstrated that the computation of the face averages $\langle N\ \frac{s}{d} \rangle\ i + \frac{1}{2}e^d$ can be reduced to integrals over cell edges. Moreover, assuming that the edge integrals are performed with the same quadratures wherever they appear,

$$\sum_{d=1}^{3} \sum_{\pm=+,-} \pm \int_{A^{\pm}_d} N^s_d dA_{\boldsymbol{\xi}} = 0, \tag{16}$$

which guarantees the freestream property that the divergence of a constant vector field computed by (12) is identically zero.

4. Newton-Krylov solution of the gyrokinetic poisson equation with adiabatic electrons

Although the model (1)-(6) can describe the evolution of electrons just as well as ions, it is often of interest to use a simpler electron model. This is especially the case when performing an explicit time integration, whose time step size would otherwise suffer from the stability restriction resulting from the small electron mass. One such model is the Boltzmann relation

$$n_e = \frac{\langle \sum_i Z_i \bar{n}_i \rangle_j}{\langle \exp(\Phi/T_e) \rangle_j} \exp(\Phi/T_e), \tag{17}$$

where $<\mu> \equiv \sum_k w_{j,k^u}\ r_j, \theta_k)$ denotes the average of μ over the j-th flux surface. Boltzmann relations such as (17) are obtained by integrating the component of the momentum equation in the magnetic field direction of a fluid model neglecting inertial terms; the resulting constant of integration appears as a prefactor that can be chosen to impose an addition constraint. In (17), the prefactor is chosen to maintain charge neutrality on (closed) flux surfaces, which is justified by the fast motion of electrons along field lines.

Because the use of (17) in (8) results in a nonlinear equation, we use a Newton iteration to solve for Φ. In each Newton iteration, a linear solve is performed whose coefficient matrix is the Jacobian $J = J_1 + J_2$, where J_1 is the gyrokinetic Poisson operator and the contribution resulting from (17) is

$$J_2 \equiv \mathrm{diag}_j \left[\frac{\langle \sum_i Z_i \bar{n}_i \rangle_j}{T_e} D_j(\Phi)(I - e^T w_j D_j(\Phi)) \right], \tag{18}$$

where

$$D_j(\Phi) \equiv \frac{\mathrm{diag}_k(\exp(\Phi_{j,k})T_e)}{\langle \exp(\Phi/T_e)\rangle_j}, \tag{19}$$

$$e \equiv (1, 1, ..., 1), \tag{20}$$

$$w \equiv (w_{j,1}, w_{j,2}, ..., w_{j,N}). \tag{21}$$

For domains whose radial width spans many gyroradii, the Jacobian term J_2 dominates J_1. However, J_2 is singular, with a null space comprised of vectors corresponding to functions that are constant on flux surfaces. This implies that the use of a Krylov method to solve the Jacobian system must be modified to avoid the near null space. In the implementation described in the next section, we employ a Bi-Conjugate Gradient Stabilized (BiCGStab) iteration with a two-step preconditioner. The first step projects the current residual onto flux surface averages and applies a tridiagonal solve to obtain a correction that is a function of radius only. The second step of the preconditioner handles the complementary piece of the residual, using a few iterations of a multigrid preconditioned conjugate gradient iteration.

5. Implementation

The algorithms summarized above provide the foundation of a new code named COGENT (COntinuum Gyrokinetic Edge New Technology) for solving the gyrokinetic Vlasov-Poisson system (1)-(6) and (8). The fourth-order, finite-volume, mapped-grid spatial discretization described in Section 3 is combined with a fourth-order Runge Kutta (RK4) time integration. In each function evaluation, the gyrokinetic Poisson equation is solved using the RK4-predicted distribution functions, yielding the electric field needed to compute the phase space velocities (2)-(6).

COGENT is built upon the Chombo library [5] under development by the SciDAC APDEC project to facilitate the creation of structured adaptive mesh refinement (AMR) applications. Although COGENT does not currently utilize Chombo's AMR capabilities, a future development path is nevertheless provided. Chombo provides support for the mapped grid formalism described in Section 3. This includes the construction of discrete metric factors from a user-specified mapping and the computation of fourth-order face-averaged fluxes via (15). COGENT also utilizes Chombo's data containers for mesh-dependent quantities distributed over processors. Such quantities are functions of configuration space (e.g., potential) or phase space (e.g. distribution functions), each of which can be domain decomposed independently. This implies the need for injection and projection operators between configuration/velocity and phase space, which have also been developed in COGENT.

6. Numerical example: Geodesic acoustic modes

As a test of the accuracy of our discretization and code, we consider the calculation of geodesic acoustic modes (GAMs), which are eigenmodes of the gyrokinetic Vlasov-Poisson system [15]. More specifically, a GAM is an ion acoustic wave driven by components of the **E** × **B** force related to the geodesic curvature of field lines on flux surfaces. The radial component of **E** induces a perpendicular flow which, to maintain continuity, results in a density perturbation. The density perturbation generates a current that in turn transports charge across the flux surfaces acting to reverse **E**. The interaction of these processes results in a damped wave. Figure 3 contains plots of ion density (top row) and potential (bottom row) computed by COGENT in the vicinity of an equilibrium flux surface [12] for a sample GAM problem. The times shown are before (left), during (middle) and after (right) one of the direction reversals of the primarily radial electric field. Figure 4 shows the damped oscillation of the potential at a probe located at the point indicated by the "X" in the bottom right panel of Figure 3.

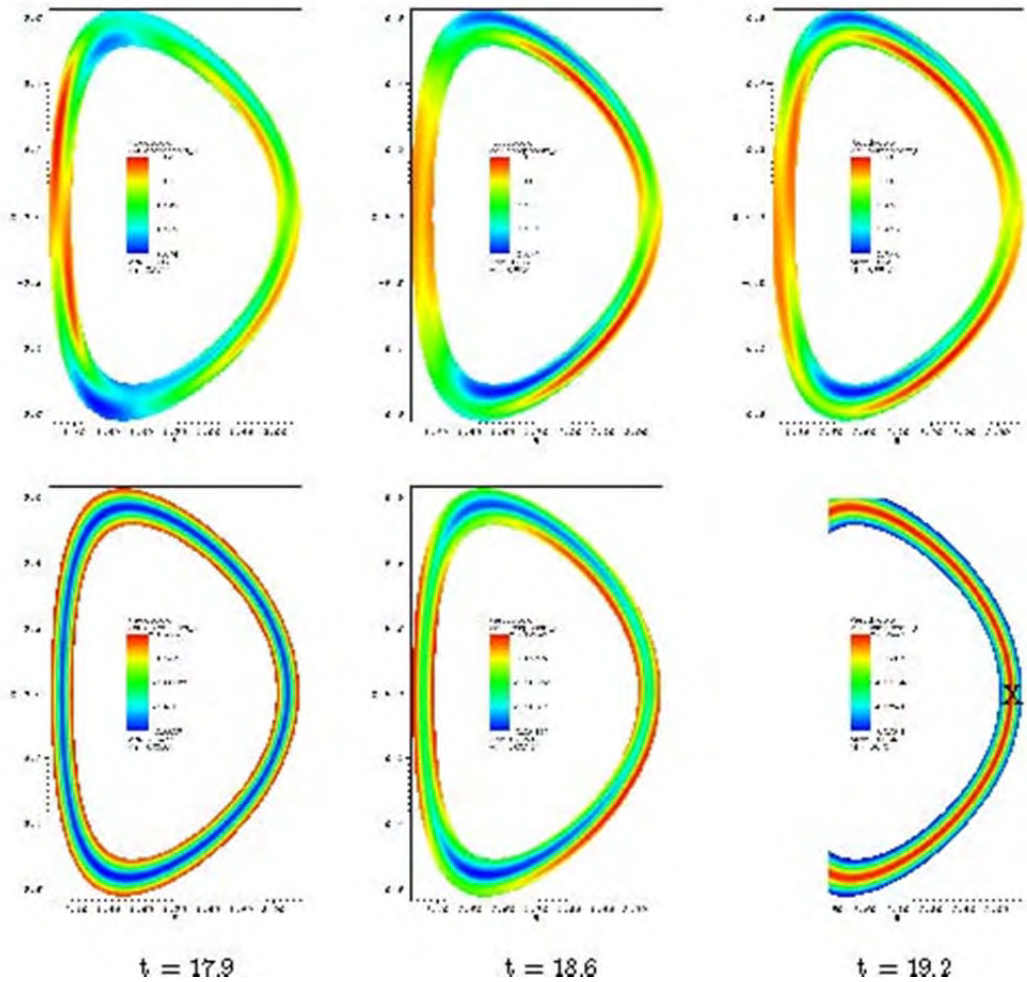


Figure 3. Ion density (top row) and potential (bottom row) in the vicinity of an equilibrium flux surface. The columns correspond to times prior to, during and after a reversal of the electric field.

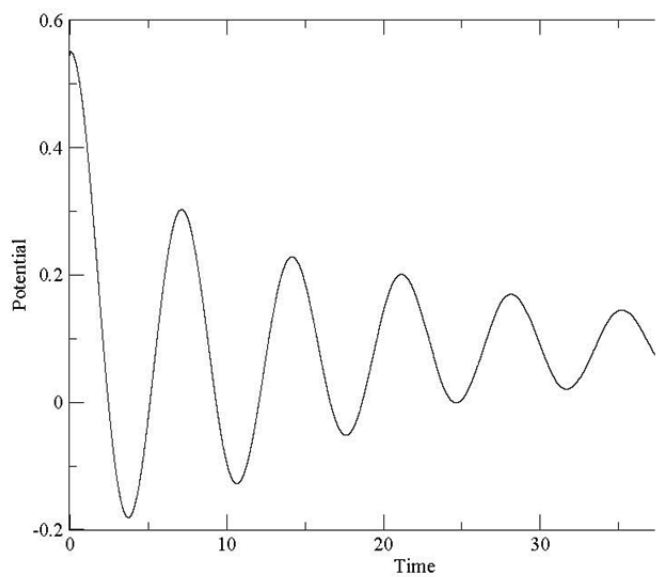


Figure 4. Potential at probe.

In the case of a circular geometry, the GAM frequency and damping rate can be predicted by a dispersion analysis of a linear gyrokinetic equation. In [7], frequencies and damping rates are obtained as functions of the ion thermal velocity v_{ti}, major radius **R**, ratio of electron to ion temperature $\tau = T_e/T_i$, and field line safety factor q. The safety factor q is the number of times a field line winds around the toroidal direction of the torus for each time it winds around the poloidal direction. It thus defines the pitch of a field line, which is related to its geodesic curvature. For large q, the frequency ω and damping rate γ are obtained asymptotically as [7]

$$\omega^2 = \left(\frac{7}{4} + \tau\right) \frac{v_{ti}^2}{R^2} \left[1 + \frac{46 + 32\tau + 8\tau^2}{(7 + 4\tau^2)q^2}\right], \tag{22}$$

$$\gamma = -i\frac{\pi^{1/2}}{2} \frac{v_{ti}}{R} \frac{(R\omega/v_{ti})^6}{(7/4 + \tau)} q^5 \exp\left[-\left(\frac{qR\omega}{v_{ti}}\right)^2\right]. \tag{23}$$

For small q, frequencies and damping rates cannot be expressed in such an analytic form and must instead be obtained by solving for the roots of a nonlinear system of equations. The system includes the resonant responses corresponding to an arbitrary number of toroidal modes.

In Figure 5, the damping rates and frequencies predicted by COGENT are plotted against curves obtained from evaluation of the dispersion model of [7] as reported in

[16] for $\tau = 0$. It is not possible to set the electron temperature to zero in COGENT, so a small value yielding $\tau = 0.1$ is used instead. The theory result includes harmonic resonances through tenth order. Figure 5 also shows the computed rates and frequencies for $\tau = 1$ compared with a curve obtained by evaluating the model of [7]. The latter includes resonances through third order, which is sufficient to converge the result for the range of q values shown.

To investigate the accuracy of the GAM results, we performed a convergence study using a sequence of refined grids and Richardson extrapolation. Beginning with an $8 \times 32 \times 32 \times 8 (r \times \theta \times v_{||} \times \mu)$ grid, we generated 3 more grids by refining all phase space dimensions by a factor of two. We then integrated the GK Vlasov-Poisson system to a fixed time on each grid and computed the differences $d_N \equiv ||n_N - n_{N-1}||$, where n_N is the ion density on the Nth refinement level, $1 \le N \le 4$, and the norm is the L1, L2 or Max norm. The convergence rates and errors for $N = 3$ and 4 are then estimated by $\rho_N \equiv \log(d_N/d_{N-1})$ and $\epsilon_N \equiv \log(d_N)/(1 + 2^{\rho N})$, respectively. The results are shown in Table 1, where the fourth-order convergence of the error is observed.

In Figure 6, the differences $\epsilon_N \equiv ||n_N - n_4||$ are plotted for $1 \le N \le 3$, approximating the error in the solutions on the first three refined grids by assuming that the solution on the fourth refined grid is exact. The black curve (only whose slope and not vertical position is relevant) indicates fourth-order convergence.

7. Summary

A gyrokinetic code for the predictive simulation the plasma edge of a fusion reactor involves multiple components, including the accurate and efficient advection of full distribution functions in 4D and 5D phase space. We have described here our progress in the development and implementation of a particular discretization of a nonlinear, full-f gyrokinetic Vlasov-Poisson system in mapped coordinates. The approach treats the configuration and velocity components of phase space on an equal footing, using a fourth-order, finite-volume, mapped grid spatial discretization combined with a fourth-order Runge Kutta time integration. The algorithm has been implemented in a new code named COGENT, which is built on the Chombo Adaptive Mesh Refinement library being developed by the SciDAC APDEC project. In COGENT, the phase space, configuration space and velocity space grids may be decomposed independently, allowing degrees of freedom to be deployed most efficiently. The algorithm and code have been tested on geodesic acoustic mode problems in a standard equilibrium magnetic geometry by comparing damping rates and frequencies with those predicted by theoretical dispersion analyses. Convergence tests were also performed to verify the expected increase in accuracy as the phase space grid was refined.

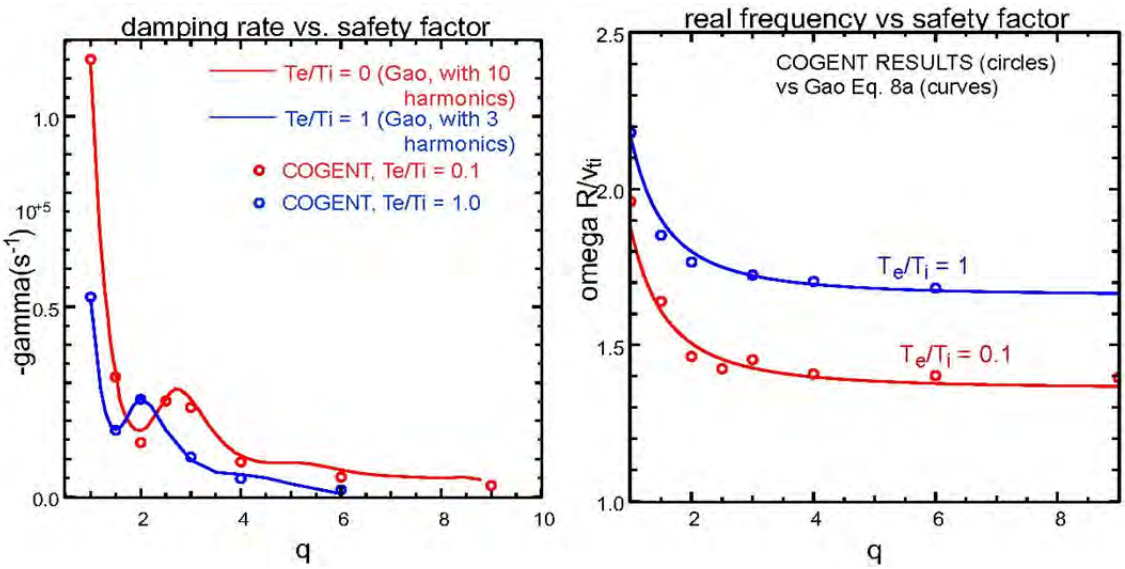


Figure 5. GAM damping rate (left) and frequency (right) as a function of safety factor.

Table 1. Estimated convergence rates and extrapolated errors for a sequence of refined grids

Grid refinement level N	Grid ($r \times \theta \times v_{\|\|} \times \mu$)	Estimated density error conv. rate ρ_N	Richardson extrapolated density error ϵ_N
1	$8 \times 32 \times 32 \times 8$		
2	$16 \times 64 \times 64 \times 16$		
3	$32 \times 128 \times 128 \times 32$	3.8 (L1)	3.37×10^{-7} (L1)
		3.8 (L2)	6.03×10^{-7} (L2)
		4.1 (Max)	1.95×10^{-4} (Max)
4	$64 \times 256 \times 256 \times 64$	4.2 (L1)	1.40×10^{-8} (L1)
		4.1 (L2)	2.95×10^{-7} (L2)
		3.6 (Max)	1.69×10^{-5} (Max)

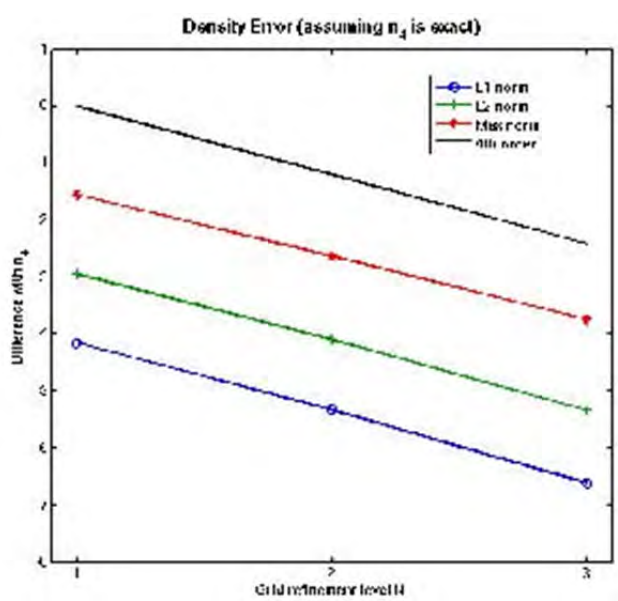


Figure 6. Error as a function of refinement level, assuming the solution on the finest grid is exact. The black curve indicates a fourth-order convergence rate.

References

[1] J. Candy. Gyro. http://fusion.gat.com/comp/parallel/.
[2] J. Candy and R.E. Waltz. Anomalous transport in the DIII-D tokamak matched by supercomputer simulation. *Phys. Rev. Lett.*, 91(4):045001–1–4, 2003.
[3] J. Candy and R.E. Waltz. An Eulerian gyrokinetic-Maxwell solver. *J. Comput. Phys.*, 186:545–581, 2003.
[4] P. Colella, M. R. Dorr, J. A. F. Hittinger, and D. F. Martin. High-order, finite-volume methods in mapped coordinates. *J. Comput. Phys.* Submitted. Also available as Lawrence Livermore National Laboratory report LLNL-JRNL-385708.
[5] P. Colella, D. T. Graves, T. J. Ligocki, D. F. Martin, D. Modiano, D. B. Serafini, and B. Van Straalen. Chombo Software Package for AMR Applications-Design Document. http://seesar.lbl.gov/anag/chombo.
[6] W. Dorland, F. Jenko, M. Kotschenreuther, and B.N. Rogers. Electron temperature gradient turbulence. *Phys. Rev. Lett.*, 85:5579, 2000.
[7] Z. Gao, K. Itoh, H. Sanuki, and J. Q. Dong. Eigenmode analysis of geodesic acoustic modes. *Phys. Plasmas*, 15:072511–1–072511–9, 2008.
[8] T. S. Hahm. Nonlinear gyrokinetic equations for turbulence in core transport barriers. *Phys. Plasmas*, 3(12):4658–4664, 1996.
[9] F. Jenko. Massively parallel Vlasov simulation of electromagnetic drift-wave turbulence. *Comput. Phys. Commun.*, 125:196, 2000.
[10] F. Jenko, T. Dannert, and C. Angioni. Heat and particle transport in a tokamak: Advances in nonlinear gyrokinetics. *Plasma Phys. Contr. F.*, 47:B195, 2005.
[11] M. Kotschenreuther, G. Rewoldt, and W. M. Tang. Comparison of initial value and eigenvalue codes for kinetic toroidal plasma instabilities. *Comput. Phys. Commun.*, 88(128), 1995.
[12] R. L. Miller, M. S. Chu, J. M. Greene, Y. R. Lin-Liu, and R. E. Waltz. Noncircular, finite aspect ratio, local equilibrium model. *Phys. Plasmas*, 5(4):973–978, 1998.
[13] G. D. Porter, R. Isler, J. Boedo, and T. D. Rognlien. Detailed comparison of simulated and measured plasma profiles in the scrape-off layer and edge plasma of diii-d. *Phys. Plasmas*, 7(9):3663–3680, 2000.
[14] Z. Qiu, L. Chen, and F. Zonca. Collisionless damping of short wavelength geodesic acoustic modes. *Plasma Phys. Contr. F.*, 51:012001, 2009.
[15] N. Winsor, J. L. Johnson, and J. M. Dawson. Geodesic acoustic waves in hydro-magnetic systems. *Phys. Fluids*, 11(11):2448–2450, 1968.
[16] X. Q. Xu, E. Belli, K. Bodi, J. Candy, C. S. Chang, R. H. Cohen, P. Colella, A. M. Dimits, M. R. Dorr, Z. Gao, J. A. Hittinger, S. Ko, S. Krasheninnikov, G. R. McKee, W. M. Nevins, T. D. Rognlien, P. B. Snyder, J. Suh, and M. V. Umansky. Dynamics of kinetic geodesic-acoustic modes and the radial electric field in tokamak neoclassical plasmas. *Nucl. Fusion*, 49:065023, 2009.

Multiple multiresolution representation of functions and calculus for fast computation

G. I. Fann,[1] R. J. Harrison,[1,2] J. Jia,[1] J. Hill,[1] and D. Galindo[1]

[1]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830
[2]Department of Chemistry, University of Tennessee, Knoxville, TN 37996

Email: fanngi@ornl.gov

Abstract. We describe the mathematical representations, data structure and the implementation of the numerical calculus of functions in the software environment multiresolution analysis environment for scientific simulations, MADNESS. In MADNESS, each smooth function is represented using an adaptive pseudo-spectral expansion using the multiwavelet basis to a arbitrary but finite precision. This is an extension of the capabilities of most of the existing net, mesh and spectral based methods where the discretization is based on a single adaptive mesh, or expansions.

1. Introduction

One of the appealing features of the MADNESS computational environment which enhances its usability is the high level composition which permits codes to be expressed similiar to the mathematical equations describing the physics. In order to make this possible, an adaptive representation of functions using multiresolution analysis for numerical calculus is developed to preserve the accuracy of the computation. In this paper, we focus on the computer science and implementation aspects of MADNESS. In particular, the numerical calculus for permitting arithmetic of functions using multiple resolutions is described as well as its parallel implementations to achieve user defined, arbitrary but finite, precision. We work with smooth functions and assume that they can be represented and sampled in an interval in the multiwavelet basis. Previously, the approximation of discontinuous functions was described [4].

MADNESS is designed to address the following programming and computational issues:

- raising the level of composition of scientific applications, making it faster and easier to both construct robust and correct algorithms and calculate solutions to existing and new problems,
- computing using functions and operators instead of just numbers, vectors and matrices,
- providing fast and accurate solutions to a variety of differential and integral equations in one to six dimensions (perhaps higher), and
- facilitating the use of massively parallel computational resources by a wider audience.

Using MADNESS, a code is written in terms of functions and operators, using the *C++* language and for this reason, it can be thought of as a basis-free method. There are underlying representations and approximations using bases and grids that adapt and refine automatically to satisfy the user requested precision. However, the user does not have to think about either the adaptivity or refinement until efficiency or memory become an issue.

The numerical operations can be regarded as a finite precision symbolic calculus used to express the equations. For example, the code corresponding to

$$\int_{\Omega} \frac{1}{2}\left(\frac{d\psi}{dx}\right)^2 + V(x)\psi(x)^2 d\Omega$$

in three dimensions is just

```
real_function_3d dpsidx=Dx(psi);
double result = 0.5 * inner(dpsidx, dpsidx) + inner(V(x), psi*psi);
```

where psi is a 3-D function, Dx(psi) computes the derivative of psi in the x-direction, and the method inner computes the L_2 inner product $(f,g)=\int f(x)g(x)\,dx$.

Within MADNESS both differential operators and integral operators can be applied to functions. Important and common convolution operators with physically relevant free space Green functions (e.g., Coulomb, bound-state Helmholtz, free-particle quantum propagator) are built in. The details of their implementation will be described elsewhere. Finally, MADNESS excels at obtaining high-accuracy solutions, computing efficiently in many dimensions, and using massively parallel computers. However, MADNESS is not appropriate for every application. In particular, if you have complicated boundary conditions or geometries, highly oscillatory functions, or only need low precision, then other tools might be more useful.

In the following, we describe approximation of functions using multiwavelets and explain the data structure and parallel computing approach which enable its usability. In Section 2, we review multiwavelets. In Section 3, we describe the approximation of functions and the computation of some elementary composition of functions in MADNESS. In Section 4, parallel implementation and programming issues are outlined.

2. Multiwavelets

We describe our algorithms in one dimension. Most of the description carries over to many dimensions via tensor products and by rescaling the domain. We note that in higher dimensions it is advantageous to use the non-standard representations [3] to decouple interactions between scales. We use wavelet and multiwavelet interchangeably when it is convenient and clear.

We adapt the notation of [1,2] and review some results. Fix a positive integer k and the unit interval. Let $\phi_i(x) = \sqrt{2i+1}P_i(2x-1)$ denote i-th scaling function, which is the i-th Legendre polynomial $P_i(x)$ shifted to $(0,1)$ and normalized. For $k=0$, $\phi_0(x)$ is just the Haar scaling function. This set of functions span the space of "scaling functions" V_0.

Define functions $\phi_{il}^n(x) = 2^{n/2}\phi_i(2^n x - l)$, where $n = 0,1,\ldots$ and $l = 0,\ldots,2^n-1$, the dyadical scaled and translated of the scaling functions. These functions are orthonormal on $[2^{-n}l, 2^{-n}(l+1)]$ and vanish elsewhere. Denote the span of these functions by V_n^k.

The scaling functions were constructed with the following properties:

- $L_2([0,1]) = \overline{lim_{n\to\infty} V_n^k}$
- Given an initial level 0, $V_0^k \subset V_1^k \subset V_2^k \ldots$
- Invariant under translation
- If $f \in V_n$, then $f(2x) \in V_{n+1}$
- For a given scale n, $\int \phi_{il}^n(x)\phi_{jm}^n(x)dx = \delta_{ij}\delta_{lm}$[1].

[1]The δ_{ij} is the Kronecker delta function.

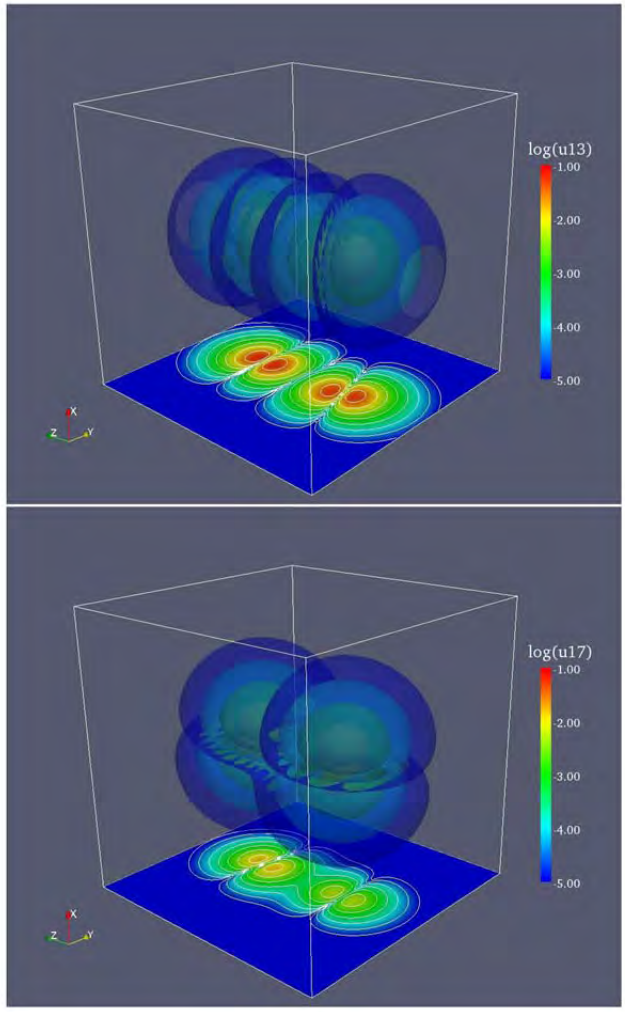


Figure 1. MADNESS has be used to compute electronic structure of molecules and nuclear structures of elements. The logarithm of the absolute value of the real part of the quasi-particle wavefunctions for nuclear density functional theory simulations [6].

The multiwavelet space W_n^k, is defined to be the orthogonal complement of V_n^k in V_{n+1}^k. In Alpert's thesis, there are two choices for the orthonormal multiwavelets, $\psi_0(x)$ in W_0^k; vanishing moments or regularity. We primarily use the multiwavelets with vanishing moments, i.e., $\int \psi_j(x)x^i dx = 0$, for $i,j = 0,\ldots,k-1$. In particular, if a function has a Taylor expansion up to order $k-1$, then the wavelet coefficients up to order $k-1$ are zero. The space W_n^k is spanned by $2^n k$ functions obtained from $\psi_0(x),\ldots,\psi_{k-1}(x)$ by dilation and translation, $\psi_{jl}^n(x) = 2^{n/2}\psi_j(2^n x - 1)$.

A key property is that Alpert's multiwavelets generate a form of multiresolution analysis [5] and are orthogonal. Let $L_2([0,1])$ represent the space of L_2-integrable functions on the unit interval with the L_2 metric, $< f,g >= \int_0^1 fgdx$.

Without loss of generality, we drop the k notation, unless necessary. Since $V_0 \subset V_1$, the basis functions for level 0 can be expressed in terms of the basis of V_1. Thus there exists constants $h_{ij}^0, h_{ij}^1, g_{ij}^0, g_{ij}^1$ such that

$$\phi_i(x) = \sqrt{2}\sum_{j=0}^{k-1} (h_{ij}^0 \phi_j(2x) + h_{ij}^1 \phi_j(2x-1)),$$

and

$$\psi_i(x) = \sqrt{2}\sum_{j=0}^{k-1} (g_{ij}^0 \phi_j(2x) + g_{ij}^1 \phi_j(2x-1)).$$

The key property of multiresolution analysis (MRA) is the separation of the behavior of functions and operators at different length scales. Central to this concept is the telescoping series which exactly represents the basis at a finest scale n in terms of basis from level 0 with successive corrections at successively finer scales,

$$V_n = V_0 + (V_1 - V_0) + \ldots + (V_n - V_{n-1}).$$

If the function is sufficiently smooth and regular in some region to be represented at the desired precision at some level, then the differences at finer scales will be negligibly small.

Each definition of multiwavelet basis and the scaling function for each level n gives a dyadic subdivision of the unit interval which is naturally represented as a dyadic tree.

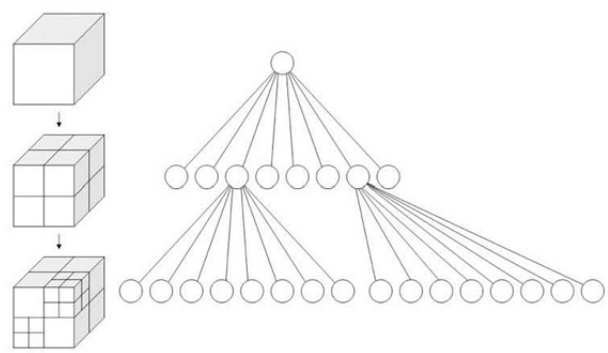


Figure 2. The multiresolution representation of a function using multiwavelets is naturally represented in a tree structure. A 3-D function is shown which corresponds to an octree (Octree picture is from WhiteTimberwolf's contribution at Wikipedia).

3. Numerical Calculus in MADNESS

In order to perform function calculus in a user friendly fashion, arithmetic operations of functions and functions of functions must also be defined and computed both efficiently and in a fast manner in the multiwavelet representation without resorting to multiple transformations and costly quadratures for resampling. The orthonormality property of the multiwavelet basis make these computations fast and efficient.

We describe the computation of a few of these representations below.

3.1. Approximation of Smooth Functions

Let $f \in L_2([0,1])$ be a smooth function. Its projection, $\hat{f} \in V_n$ can be written as $\hat{f}(x) = \sum_{l=0}^{2^n-1} \sum_{j=0}^{k-1} s_{jl}^n \phi_{jl}^n(x)$. The coefficients s_{jl}^n are computed as $s_{jl}^n = \int_{2^n l}^{2^n(l+1)} f(x)\phi_{jl}^n(x)dx$. The function expanded in the multiwavelet basis is given as

$$f(x) = \sum_{j=0}^{k-1} s_{j,0}^{0}\phi_j(x) + \sum_{j=0}^{k-1}\sum_{m=0}^{n-1}\sum_{l=0}^{2^m-1} d_{jl}^{m}\psi_{jl}^{m}(x),$$

with the multiwavelet coefficients computed as $d_{jl}^{m} = \int_{2^{-m}l}^{2^{-m}(l+1)} f(x)\psi_{jl}^{m}(x)dx$. Since the basis are piecewise polynomial an adaptive Gauss-Legendre quadrature is used.

We use the term "compression" to mean multiwavelet compression, analogous to wavelet compression of an image. In many examples an image or a signal (denoted by f) can be written as $f = u + v$ where u is a structured component and v is a noisy component. The structure part u can often be represented using very few coefficients (in a compressed form).

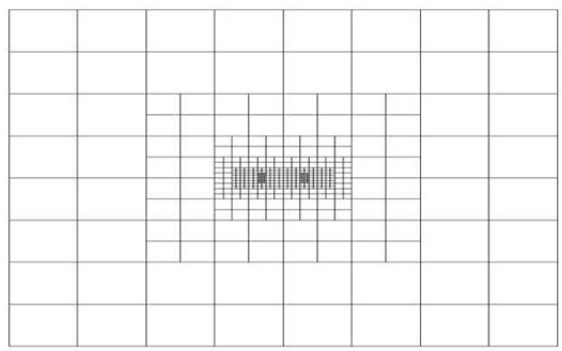


Figure 3. A two-dimensional slice of the three-dimensional adaptive refinement of cubes shows where significant scaling and multiwavelet coefficients are required for the representation of the singular potential corresponding to the nuclei of the Hydrogen molecule.

3.2. Truncation Criteria

Discarding small differences coefficients in the multiwavelet expansion while maintaining precision is crucial to speed and sparsity, and drives the adaptive refinement. Recall that for smooth functions the level of the multiwavelet expansion depends on the user given error criteria and is dependent, dyadically, on the averaged value of the basis functions and dies off with respect to level of adaptive refinement and the box sizes.

Different truncation criteria are used in MADNESS in the multiwavelet representation. Let d_l^n denote the tensor of coefficients d_{il}^n for level n, leaf node l. The norm $||d_l^n||_2$ denotes the norm $\sum_i \sqrt{|d_{il}^n|^2}$ of d_l^n.

Mode 0: The difference coefficients of the leaf nodes are neglected according to

$$\sum_i \sqrt{|d_{il}^n|^2} \leq \varepsilon.$$

This truncation is appropriate for most calculations where the functions have deep levels of refinements (such as those near singularities, steep gradients or discontinuities).

Mode 1: Mode 1 is appropriate when an accurate representation of both the function and its derivative is required. Difference coefficients of leaf nodes are neglected according to

$$||d_l^n|| \leq min(1, c * 2^{-n})\varepsilon$$

where c is a constant depending on the simulation domain size before rescaling to the unit interval.

Mode 2: Mode 2 is appropriate only for smooth functions with nearly uniform level of refinement for the entire domain. The multiwavelet difference coefficients are neglected according to

$$||d_l^n|| \leq 2^{-nd/2}\varepsilon$$

where d is the dimension of the approximation. In this mode, the error between the representation between the levels $n-1$ and n is within ε, $||f^n - f^{n-1}||_2 \leq \varepsilon$.

3.3. Adaptive Refinement

After projection has been performed at boxes number $2l$ and $2l+1$ at level n, the scaling function coefficients may be tranformed or filtered to produce multiwavelet coefficients at box l at level $n-1$. If the desired truncation is not satisfied, the process is repeated at the child boxes $4l, 4l+1, 4l+2, 4l+3$ at level $n+1$. Otherwise, the computed coefficients are inserted at level n.

3.4. Evaluation

Once a function has been represented at level n with the desired approximation. The function can be evaluated at different points by recurring down the dyadic tree to find the appropriate box which contains the point, evaluating the basis functions and summing over its products with the coefficients.

3.5. Products

We briefly describe the procedure for multiplication of multiwavelet functions represented in an interval [2]. Consider two functions f and g approximated in the multiwavelet basis, each represented on the interval by k-term interpolating scaling polynomial expansions. The interpolating scaling functions $\phi(x)$ constructed from the Legendre interpolating polynomial at the quadrature points is related to the scaling functions $\phi(x)$ by

$$\phi_j(x) = \sqrt{w_j}\sum_{i=0}^{k-1} \phi_i(x_j)\phi_i(x)$$

where $x_0, \ldots x_{k-1}$ denote the roots of the Legendre polynomail $P_k(2x-1)$ and its associated quadrature weights $w_0, \ldots w_{k-1}$. There is a fast transform to and from the multiwavelet scaling functions and the interpolating scaling functions.

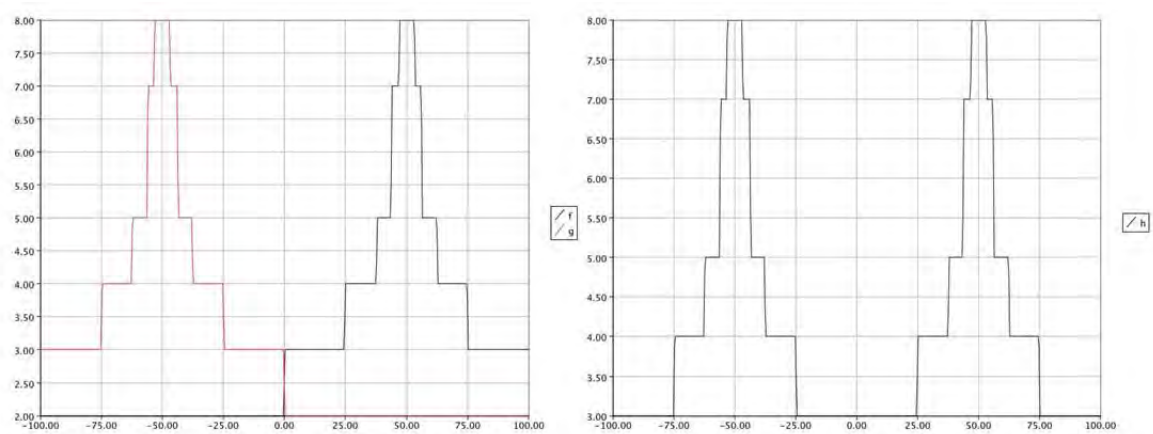


Figure 4. The adaptive structure of two functions $f(x) = 1 + e^{(-x+50)^2}$ and $g(x) = 1 + e^{(-x-50)^2}$ and their product $h = fg$ is shown. The vertical axis indicates the level n of the multiwavelet expansion that is used in the adaptive discontinuous expansion.

We obtain the product $f \cdot g$ as follows. Let $f(x) = \sum_{j=0}^{k-1} \sum_{l=0}^{2^n-1} a_{jl}\phi_{jl}^n(x)$ and $g(x) = \sum_{j=0}^{k-1} \sum_{l=0}^{2^n-1} b_{jl}\phi_{jl}^n(x)$, with possibly two different n. Then:

- Refine the functions f and g so that the representations of the two functions are at the same level.
- Now refine the representation of f and g by dividing the subinterval on the finest level into two, to allow sufficient resolution to represent the product.
- The coefficients of the product of representations $h(x) = f(x)g(x) = \sum_{j=0}^{k-1} c_{jl}\phi_{jl}^n(x)$ on a subinterval at level n is $c_j = (2^{n/2})\frac{a_j b_j}{\sqrt{w_j}}$, where w_j are the Gaussian quadrature weights.
- The threshholding is applied. The coefficients for the two finest levels n and $n+1$ are compared. If the coefficiencts at level $n+1$ are below the truncation threshold, those coefficients are deleted.

The case of squaring a function is special and can be performed faster since it is an in-place operation. In some scenarios, it can be used to multiply two functions. This follows from Beylkin's observation that $f \cdot g = \frac{1}{4}((f+g)^2 - (f-g)^2)$.

In MADNESS, a slight different approach is used. Let us describe the point-wise multiplication since this is fundamental. The products of two functions is performed in the scaling function basis. Intuitively, each function is transformed to values at the quadrature points, multiply the values and then transformed back. In practice and in robust implementation there are three complicating issues.

First, the product cannot be exactly represented in the polynomial basis. The product of two polynomials of order $k-1$ produces a polynomial of order $2k-2$. Beylkin makes a nice analogy to the product of two functions sampled on a grid – the product can be exactly formed on a grid with double the resolution. This is not exact for polynomails but the expected error is reduced by a factor 2^{-k} where k is the number of multiwavelets.

In MADNESS, an option to automatically refine and form the product at a lower level is given. This is performed by estimating the norm of each of the leaf boxes which is a products of the Frobenius norm of the two functions.

Second, the functions may have different levels of adaptive refinement. The two options are to compute the function with coefficients at a coarser-scale directly on the grid required for the finer-scale function, or to refine the function down to same level, which is what we previously choose to do. However, this leaves the tree with scaling function coefficients at multiple levels that must be cleaned up after the operation. Since it essential (for efficient parallel computation) to perform multiple operations at a time on a function, having it in an inconsistent state makes things complicated. If all we wanted to do were perform other multiplication operations, there would be no problem; however this seems to be an unnecessary restriction on the user. It is also faster to perform the direct evaluation and this is what has been implemented.

Third, the above does not use sparsity or smoothness in the functions and does not compute the result to a finite precision. For instance, if two functions do not overlap their product is zero but the above algorithm will compute at the finest level of the two functions doing a lot of work to evaluate small numbers that will be discarded upon truncation. Eliminating this overhead is crucial for reduced scaling in electronic structure calculations. At some scale we can write each function in the domain in terms of its usual scaling function approximation at that level s and the correction/differences d from all finer scales. The error in the product of two such functions is then the sum of the products of the approximation of the scaling function coefficients and multiwavelet difference coefficients for f and g.

Note again that while the scaling function coefficients are as used everywhere else in this document, the difference multiwavelet part of the expansion is the sum of corrections at all finer scales. Thus, by computing the scaling function coefficients at all levels of the tree and summing the norm of the differences up the tree we can compute with controlled precision at coarser levels of the tree. The sum of

the norm of differences can also be computed by summing the norm of the scaling function coefficients from the finest level and subtracting the local value.

3.6. Quotients and Rational Functions

Similiar to products we describe the procedure for representating rational functions $f(x)/g(x)$. In this case we have to find the coefficients of $h(x)$ such that $f(x) = h(x)g(x)$. From the procedures for products, at level n, $a_j = (2^{n/2})\frac{c_j b_j}{\sqrt{w_j}}$. Thus, $c_j = 2^{-n/2}\sqrt{w_j}a_j/b_j$ with $b_j \neq 0$.

3.7. Square Root

The square root of a function can also be described. Using the products of two functions we have the coefficients for $h(x) = f^2(x)$ being represented on level n as $c_j = (2^{n/2})\frac{a_j a_j}{\sqrt{w_j}}$. The coefficients of the function $f(x) = \sqrt{h(x)}$ can be represented as $\sqrt{2^{-n/2}c_j\sqrt{w_j}}$.

Computation of cubic and higher roots of functions can be performed in an analogous way.

4. Data Structure and Computational Methodologies

Multiwavelet representations are best understood in terms of 2^d-tree for d dimensions correspond to the dyadic scaling of the basis functions in the tensor product representation. Each level of the tree corresponds to a level of the refinement using the 2-scale relation in each dimension. The difficulties of the tree architecture is the serial manner in which efficient tree traversal algorithms are implemented, as the order of summation that must be performed must vary to minimize round-off or numerical summation errors. In MADNESS, the tree is mapped to a hash table. In this section we assume the reader has some knowledge of $C++$ data structure and templates as well as parallel computing.

A d-dimension function is represented as a 2^d-tree. Nodes in the tree are labeled by a tuple key (n, l) where n is the level and l is a d-vector representing the translation. Nodes are represented by instances of FunctionNode<T,d>, where T is a templated data type, that presently contains the coefficients and an indicator if this node has children. Nodes, indexed by keys, are stored in a distributed hash table that is an instance of a MADNESS container class. This container uses a two-level hash to first map a key to the processes (referred to as the owner) in which the data resides, and then into a local instance of either a GNU hash_map or a TR1 unordered_map. Since it is always possible to compute the key of a parent, child, or neighbor we never actually store (global) pointers to parents or children. Indeed, a major aim of the MADNESS runtime environment is to replace the cumbersome partitioned global address space (global pointer is process id + local pointer) with multiple global name spaces. Each new container provides a new name space. Using names rather than pointers permits the application to be written using domain-specific concepts rather than a rigid linear model of memory.

Data common to all instances of a function of a given dimension (d) and data type (T, e.g., float, double, float_complex, double_complex) are gathered together into FunctionCommonData<T,d> of which one instance is generated per wavelet order (k). An instance of the common data is shared read-only between all instances of functions of that data type, dimension and wavelet order. In addition to reducing the memory usage of the code, sharing the common data greatly speeds the instantiation of new functions which is replicated on every processor. In order to facilitate shallow copy and assignment semantics and to make empty functions inexpensive to instantiate, a multi-resolution function, which is an instance of Function<T,d> contains only a shared pointer to the actual implementation which is an instance of FunctionImpl<T,d>. Uninitialized functions (obtained from the default constructor) contain a zero pointer. Only a non-default constructor or assignment actually instantiate the implementation. The main function class forwards nearly all methods to the underlying implementation. The implementation primarily contains a reference to the common data, the distributed container storing the tree, little bits of other state (e.g., a flag indicating the compression status) and several methods.

Default values for all functions of a given dimension are stored in an instance of FunctionDefaults<d>. These may be modified to change the default values for subsequently created functions. Functions have many options and parameters and thus we have made it easy to specify options and selectively override defaults. Since $C++$ does not provide named arguments (i.e., arguments with defaults that may be specified in any order rather than relying on position to identify an argument) we adopt the named-parameter idiom. The main constructor for Function<T,d> takes an instance of FunctionFactory<T,d> as its sole argument. The methods of FunctionFactory<T,d> enable setting of options and parameters and each returns a reference to the object to permit chaining of methods.

Most memory is reference counted using Boost-like shared pointers. An instance of SharedPointer<T>, which wraps a pointer of type T*, is used to wrap memory obtained from the $C++$ new operator. The exceptions are where management of the memory is immediately given to a low-level interface. Shared-pointers may be safely copied and used with no fear of using a stale pointer. When the last copy of a shared pointer is destroyed the underlying pointer is freed. With this mode of memory management there is never any need to use the $C++$ delete operator and most classes do not even need a destructor.

4.1. Maintaining a consistent state of the 2^d-tree

The function implementation provides a method verify_tree() that checks connectivity and consistency of the compression state, presence of coefficients, child flags, etc, as described below.

A node in the tree is labeled by the tuple (n, l) key and presently stores the coefficients and a flag indicating if the node has children. In 1D, the keys of children are readily computed as $(n+1, 2l)$ and $(n+1, 2l+1)$. In many dimensions it is most convenient to use the KeyChildIterator class. In the reconstructed form (i.e. the scaling function basis rather than the multiwavelet basis), a tree has coefficients (held in kd tensors) only at the lowest level. All interior nodes will have no coefficients and will have children. All leaf nodes will have coefficients and will not have children.

In the compressed form (wavelet basis), a tree has coefficients (in $(2k)d$ tensors) at all levels. The scaling function block of the coefficients is zero except at the coarsest level. Logically, this tree is one level shallower than the reconstructed tree since the scaling function coefficients at the leaves are represented by the difference coefficients on the next coarsest level. However, to simplify the logic in compress and reconstruct and to maintain consistency with the non-standard compressed form (see below), we do not delete the leaf nodes from the reconstructed tree. Thus, the compressed tree has the same set of nodes as the reconstructed tree with all interior nodes having coefficients and children, and all leaf nodes having no coefficients and no children.

In the non-standard compressed form (redundant basis), we keep the scaling function coefficients at all levels and the wavelet coefficients for all interior nodes. Thus, the nonstandard compressed tree has the same set of nodes as for the other two forms but with all nodes having coefficients (a $(2k)d$ tensor for interior nodes and a kd tensor for leaf nodes).

To keep complexity to a minimum we don't want to introduce special states of the tree or of nodes, thus all operations restore the tree to a standard state by their completion.

Truncation is applied to a tree in compressed form and discards small coefficients that are logically leaf nodes. Logically, because in the stored tree we still have the empty nodes that used to hold the scaling coefficients. For a node to be eligible for truncation it must have only empty children. Thus, truncation proceeds as follows. We initially recur down the tree and for each node spawn a task that takes as arguments futures indicating if each of its children have coefficients. Leaf nodes, by definition, have no children and no coefficients and immediately return their status. Once a task has all information about the children it can execute. If any children have coefficients a node cannot truncate and can immediately return its status. Otherwise, it must test the size of its own coefficients. If it decides to truncate, it must clear its own coefficients, delete all of its children, and set its has_children flag to false. Finally, it can return its own status.

Adding (subtracting) two functions is performed in the wavelet basis. If the trees have the same support (level of refinement) we only have to add the coefficients. If the trees differ, then in addition to

adding the coefficients we must also maintain the has_children flag of the new tree to produce the union of the two input trees. To permit functions with different processor maps to be added efficiently, we loop over local data in one function and send them to nodes of the other for addition. Sending a message to a non-existent node causes it to be created.

Acknowledgements
This work was supported by the Scientific Discovery through Advanced Computing (SciDAC) program of the U.S. Department of Energy, Office of Science, the Division of Basic Energy Sciences, and the Office of Advanced Scientific Computing Research, Program in Applied Mathematics and through the SciDAC Scientific Application Pilot Program under contract number DE-AC05-00OR22725 with UT-Battelle, LLC (Oak Ridge National Laboratory). It is also supported under the UNEDF SciDAC Collaboration. Computational resources were provided by the National Center for Computational Sciences at Oak Ridge.

References

[1] B. Alpert. A class of bases in L^2 for the sparse represetntation of integral operators. *SIAM J. Math. Anal.*, 24(1):246–262, 1993.
[2] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi, Adaptive solution of partial differential equations in multiwavelet bases. *J. Comput. Phys.*, 182(1):149–190, 2002.
[3] G. Beylkin, R. R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure and Appl. Math.*, 44:141–183, 1991. Yale Univ. Technical Report YALEU/DCS/RR-696, August 1989.
[4] M.E. Brewster, G.I. Fann, and Z.Y. Yang, Wavelets for electronic structure calculations. *Journal of Mathematical Chemistry*, 22(2-4):117–142, 1997.
[5] Stephane G. Mallat, Multiresolution approximations and wavelet orthonormal bases of $L^2(\boldsymbol{R})$. *Trans. Amer. Math. Soc.*, 315(1):69–87, 1989.
[6] J.C. Pei, M.V. Stoitsov, G.I. Fann, W. Nazarewicz, N. Schunck, and F.R. Xu. Deformed coordinate-space Hartree-Fock-Bogoliubov approach to weakly bound nuclei and large deformations. *Physics Review C*, 78, 064306, 2008.

Highly parallel solver for multi-scale parquet quantum modeling of strongly correlated materials*

S. Yang,[1] E. D'Azevedo,[2] H. Fotso,[1] M. Jarrell,[1] J. Liu,[1] T. Maier,[2] C. Sen,[2] and K. Tomko[3]

[1] Louisiana State University, Baton Rouge, Louisiana, 70803, USA
[2] Oak Ridge National Laboratory, Oak Ridge, Tennessee, 37831, USA
[3] Ohio Supercomputer Center, Columbus, Ohio, 43212, USA

Abstract. The parquet formalism to calculate the two-particle Green's functions of large systems requires the solution of a large, sparse, complex system of quadratic equations. If N_f Matsubara frequencies are used for a system of size N_c, and Newton's method is used to solve the nonlinear system, the Jacobian system has $O(8N_t^3)$ variables and $O(40N_t^4)$ complex entries where $N_t = N_c N_f$. For $N_t = 1024$, the nonlinear system has over 8.5 billion degrees of freedom and the sparse Jacobian will require over 351 TBytes of memory. The Jacobian is very expensive to store but the matrix-vector products can be computed directly. We are developing a highly scalable parallel solver that uses both OpenMP and MPI to exploit the multicore nodes. We present initial scalability results on the Cray XT5 that suggests the code can be scaled to solve larger problems with $N_t \geq 1024$.

1. Introduction

The two-dimensional Hubbard model has been accepted in the community as a minimum model to study the high-temperature superconducting cuprates. An adequate solution of this model is extremely challenging, especially in the interesting parameter regime of intermediate to strong coupling, where the Coulomb repulsion between electrons is of the same order or stronger than the electronic kinetic energy. Despite some recent success, our understanding of the properties of this model in this regime is therefore still limited.

The Hubbard Hamiltonian on a square lattice is written as

$$H = \sum_{\langle i,j \rangle} (t_{i,j} - \mu) c^\dagger_{i,\sigma} c_{j,\sigma} + U \sum_i n_{i,\uparrow} n_{i,\downarrow} \tag{1}$$

where $c^\dagger_{i,\sigma}$ (ci,σ) creates (destroys) an electron with spin σ on site i and $n_{i,\sigma} = c^\dagger_{i,\sigma}\ c_{i,\sigma}$ is the corresponding occupation operator. The first term describes the hopping of electrons between sites i

* This research is sponsored by the Office of Advanced Scientific Computing Research; U.S. Department of Energy. The work was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725. This project was supported by SciDAC grant DE-FC02-06ER25792. S. Yang and H. Fotso were supported by NSF PIRE grant OISE-0730290. This research used resources of the Center for Computational Sciences at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This work was supported in part by an allocation of computing time from the Ohio Supercomputer Center.

and j and the second term stands for the Coulomb repulsion U two electrons feel when residing on the same site.

Perturbation theory provides useful results only in the limits of either weak or strong coupling, but fails in the interesting regime of intermediate coupling. Methods employing resummations of Feynman diagrams to infinite order are usually biased due to the particular choice of diagrams. Exact methods such as Quantum Monte Carlo (QMC) and exact diagonalization are restricted to relatively small system size due to the computational complexity. In principle, one can imagine carrying out an analytical calculation which includes all the Feynmann diagrams. This would theoretically yield exact results on the single-particle and two-particle levels if one manages to include all relevant diagrammatic equations up to that level.

The parquet formalism is based on a two-particle self-consistent theory, where all the relevant physical conservation rules are preserved as well as the crossing symmetries at two-particle level are obeyed. It is based on the following four diagrammatic equations. To simplify the formalism, we consider systems that preserve the spin $SU(2)$ symmetry. We denote $\kappa \equiv (\mathbf{k}; i\omega_n)$ and $\nu \equiv (\mathbf{k}, i\nu_n)$ with $\omega_n = (2n + 1)\pi T$ and $\nu_n = 2n\pi T$. In the following, r is used as a general label for $r = d, m, s, t$ which label the particle-hole density and magnetic channels (d and m), and the particle-particle singlet and triplet channels (s and t). All indexes are in modulo arithmetic so that a negative index such as $-\kappa$ is equal to mod $(N_t - \kappa, N_t)$.

– The Dyson equation

$$G^{-1}(k) = G_0^{-1}(k) + \Sigma(k) \tag{2}$$

– The Schwinger-Dyson equation

$$\Sigma(k) = -(T/N)^2 \frac{U}{2} \sum\nolimits_{k',v} G(-k'+v)\, G(k')\, G(-k+v)\, F_s(v)_{-k',-k}\,, \tag{3}$$

– The Bethe Salpeter equation

$$F_r(v)_{k,k'} = \Gamma_r(v)_{k,k'} + \sum\nolimits_{k''} F_r(v)_{k,k''}\, \chi_r^0(v)_{k''}\, \Gamma_r(v)_{k'',k'} \tag{4}$$

for $r = d, m, s, t$, and with

$$\begin{aligned} \chi_r^0(v)_{k''} &= G(k'')\, G(k''+v)\,, \quad r = d, m \\ \chi_r^0(v)_{k''} &= -\tfrac{1}{2} G(k'')\, G(k''+v)\,, \quad r = s, t \end{aligned} \tag{5}$$

– The Parquet equation

$$\begin{aligned} \Gamma_d^{ph}(v)_{k,k'} &= \Lambda_d^{ph}(v)_{k,k'} - \tfrac{1}{2}(\Phi_d + 3\Phi_m)(k-k')_{\tilde{k},k'} + \tfrac{1}{2}(\Psi_s + 3\Psi_t)\left(k+\tilde{k}\right)_{-k',-\tilde{k}} \\ \Gamma_m^{ph}(v)_{k,k'} &= \Lambda_m^{ph}(v)_{k,k'} - \tfrac{1}{2}(\Phi_d - \Phi_m)(k-k')_{\tilde{k},k'} - \tfrac{1}{2}(\Psi_s - \Psi_t)\left(k+\tilde{k}\right)_{-k',-\tilde{k}} \\ \Gamma_s^{pp}(v)_{k,k'} &= \Lambda_s^{pp}(v)_{k,k'} + \tfrac{1}{2}(\Phi_d - 3\Phi_m)(k-k')_{\tilde{k},-k} + \tfrac{1}{2}(\Phi_d - 3\Phi_m)\left(k+\tilde{k}\right)_{-k',-k} \\ \Gamma_t^{pp}(v)_{k,k'} &= \Lambda_t^{pp}(v)_{k,k'} + \tfrac{1}{2}(\Phi_d + \Phi_m)(k-k')_{\tilde{k},-k} - \tfrac{1}{2}(\Phi_d + \Phi_m)\left(k+\tilde{k}\right)_{-k',-k} \end{aligned} \tag{6}$$

where $\tilde{\kappa} = \kappa' + \nu$,

$$\begin{aligned} \Phi_r(v)_{k,k'} &= \sum\nolimits_{k''} F_r(v)_{k,k''}\, \chi_r^0(v)_{k''}\, \Gamma_r(v)_{k'',k'}\,, \quad r = d, m \\ \Psi_r(v)_{k,k'} &= \sum\nolimits_{k''} F_r(v)_{k,k''}\, \chi_r^0(v)_{k''}\, \Gamma_r(v)_{k'',k'}\,, \quad r = s, t \end{aligned} \tag{7}$$

and for $r = d, m$, it is labeled as Φ; while for $r = s, t$, labeled as Ψ.

The parquet equations are derived from enforcing two-particle crossing symmetries, while the other equations are necessary for a conserving approximation. The above set of equations, however, does not form a closed loop because the fully irreducible vertices, Λ_r, are not determined internally. When Λ_r is replaced by the bare interaction, one obtains the parquet approximation. One can also imagine taking the results of an exact calculation for a small system size, using e.g., exact diagonalization or QMC methods, as an input for Λ_r in the parquet equations. In this case, one obtains the full multi-scale parquet formalism, that has been discussed in [9].

2. Algorithm

The parquet equations (2) to (7) form a system of nonlinear equations. The strength and spatial range of correlations depends on the ratio between the Coulomb repulsion U and temperature T and the nonlinear system becomes increasingly difficult to solve for large U. Each entity such as $\Gamma_t^{pp}(\nu)\ \kappa, \kappa'$ or $Fr(\nu)\ \kappa_1, \kappa_2$ can be discretized and represented as three-index $N_t \times N_t \times N_t$ arrays $\Gamma_t(\kappa_1, \kappa_2, \nu)$ and $F^r(\kappa_1, \kappa_2, \nu)$. A simple algorithm is to perform fixed-point iteration by freezing some of the unknowns to solve a subset of the nonlinear equations. Starting from an initial guess, we can solve the Bethe Salpeter equation (4) to generate $F_r(\nu)\ \kappa_1,\kappa_2$. The Φ_r and Ψ_r values can be updated using the new F_r values using (7). The new $\Gamma_r(\nu)\ \kappa, \kappa'$ values are computed by the Parquet equations (6). One can iteratively solve (4) and (6) until convergence, then update the Dyson equation and self energy in (2) and (3). Although this method is efficient and simple to implement, the iterations may become unstable for large values of $U \geq 18$.

A careful examination of the system shows that the Bethe Salpeter equations can be written as

$$\begin{aligned} F_r(v)_{k,k'} &= \Gamma_r(v)_{k,k'} + \Phi_r(v)_{k,k'}, \quad r = d, m \\ F_r(v)_{k,k'} &= \Gamma_r(v)_{k,k'} + \Psi_r(v)_{k,k'}, \quad r = s, t \end{aligned} \tag{8}$$

so that the Bethe Salpeter equations and the Parquet equations can be written as linear expressions in F_r, Γ_r, Φ_r, and Ψ_r. The nonlinearity comes mainly from the Φ_r and Ψ_r variables that are products of just $F_r(\nu)$ and $\Gamma_r(\nu)$; therefore (4) to (7) form a complex system of affine quadratic equations.

The literature is sparse on methods for solving a general system of quadratic equations. The solution of a system of quadratic equations using Newton's method has been analyzed in [6]. It shows that multiple solutions are possible and the Jacobian matrix may be exactly singular if it is evaluated at a midpoint of two solutions. Cohen and Tomasi [3] have considered the solution of a special case of a system of homogeneous bilinear equations. Their results show the problem is related to solving the generalized eigenvalue problem. Bouaricha and Schnabel [7,1,2] have considered an extension of the Newton's method to solve $F(x) = 0$ by including a low rank tensor approximation of higher derivatives

$$M(x_c + d) = F(x_c) + F'(x_c)d + \frac{1}{2}T_c dd, \quad F : \mathbb{R}^n \to \mathbb{R}^m\,, \tag{9}$$

where T_c is a three index tensor object formed by interpolation of past Newton steps. The tensor T_c is not the second derivative of $F(x)$ but is chosen to be a sum of p rank-one tensor objects. Equation (9) may be viewed as a particular system of quadratic equations. Ultimately, Newton's method or Levenberg-Marquardt method is used to solve (9) in a least squares sense as a smaller system of p

quadratic equations. The analysis of quadratic matrix equation $AX^2 + BX + C = 0$ in $n \times n$ matrices and the connection to quadratic eigenvalue problem $(\lambda^2 A + \lambda B + C)x = 0$ have been considered by Higham and Kim [4]. The authors used Newton's method with exact line searches [5] to solve the quadratic matrix equations.

We have developed a parallel solver using Newton's method with line search to solve the complex system of biaffine quadratic equations. The unknowns are eight $N_t \times N_t \times N_t$ arrays for F_d, F_m, F_s, F_t, Γd, Γ_m, Γ_s, and Γ_t. The complicated vertex rotations to enforce crossing symmetries in the Parquet equations (6) can be viewed as permutation operations on a long vector of length N_t^3. The permutation is implemented using the Message Passing Interface (MPI) all-to-all communication primitive. This operation places high demands on the communication network and is one of the most time consuming kernels. The sparse Jacobian matrix of the bilinear quadratic equations can be analytically computed but it has $O(40N_t^4)$ nonzeros. For problems of interest $N_t \geq 256$, the sparse Jacobian matrix is very costly to store in memory. For example, if $N_t = 1024$ then there are nearly 8.6 billion degrees of freedom and explicit storage of the sparse Jacobian still requires over 351 TBytes of memory and for a larger case of $N_t = 1280$, there are over 16.7 billion degrees of freedom and storage of the Jacobian requires over 858 TBytes of memory! For this application, we have found that computing the matrix-vector operation by finite differences introduces an unacceptably high error due to numerical cancellation. Instead the action of matrix-vector multiply is computed analytically without explicit formation of the Jacobian. The Φ and Ψ expressions in (7) are simple products of F_r and Γ_r. The interaction of derivatives of Φ_r or Ψ_r with respect to F_r or Γ_r can be computed as *dense complex* matrix products of N_t by N_t matrices. The entries in the Jacobian matrix consist of terms [derived from (7)] such as

$$\frac{\partial \Phi_r(v)_{k,k'}}{\partial \Gamma_r(v)_{k,k''}} = \chi_r^0(v)_{k''}\Gamma_r(v)_{k'',k'} \text{ or } \frac{\partial \Phi_r(v)_{k,k'}}{\partial \Gamma_r(v)_{k'',k'}} = F_r(v)_{k,k''}\chi_r^0(v)_{k''} \ . \tag{10}$$

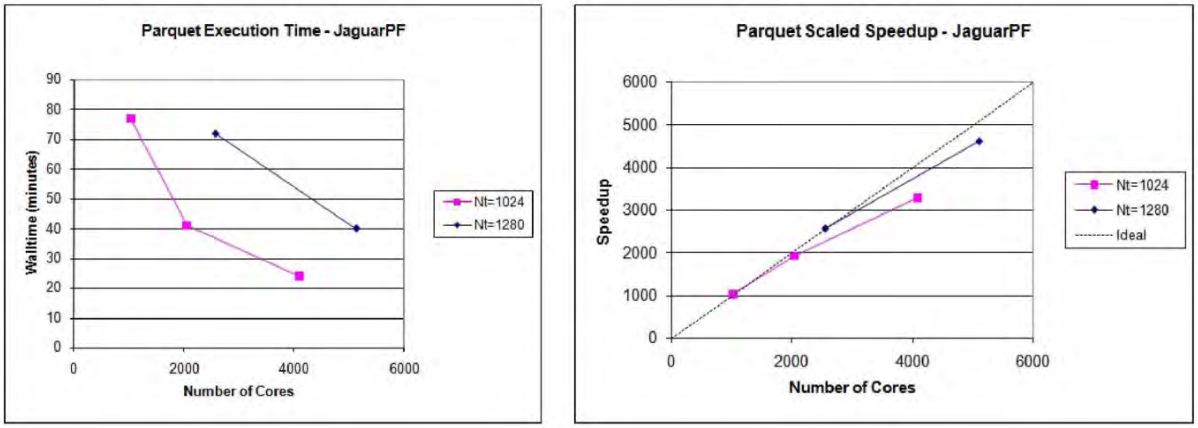
The large sparse Jacobian system is solved using the BICGSTAB(L=2) [8] Krylov iterative method† without preconditioning. Preconditioning using simple Jacobian diagonal scaling is not effective for this problem.

3. Numerical Experiments

The parquet code has a robust multilevel parallel implementation and has been ported to several parallel machines, including the International Business Machine (IBM) Opteron cluster (Glenn) at the Ohio Supercomputer Center (OSC), the Sun Constellation Linux cluster (Ranger) at the Texas Advanced Computing Center (TACC), the IBM BlueGene/P (Eugene), and Cray XT4/5 (Jaguarpf) at the National Center for Computational Sciences (NCCS) at the Oak Ridge National Laboratory (ORNL).

We present performance for the 4 × 4 Hubbard cluster model ($N_c = 16$), with parameters, $U = 0.3$, $T = 0.05$, $t = 0.25$ for two cases, using $N_f = 64$ and $N_f = 80$ Matsubara frequencies for problem sizes $N_t = 1024$ and $N_t = 1028$ respectively. For the $N_t = 1024$ case, the number of MPI processes was varies from 256 to 1024, with 4 OpenMP threads per MPI process, using a maximum of 4096 cores. The $N_t = 1280$ case was run with 640 and 1280 MPI processes with 4 threads per process using a maximum of 5120 cores. The plots in Figure 1 show the code performs well despite the need for a large "all-to-all" exchange for tensor rotations.

† The code for bicgstab2 is available at http://www.math.uu.nl/people/vorst/zbcg2.f90



(a) Execution times (b) Scaled speedup

Figure 1. Strong Scaling plots for the Parquet Solver base on Jaguarpf measurements (Left) Execution times for two problem sizes, on core counts from 1024 to 5120. (Right) Scaled speedup plot for the Parquet code, speedup for each problem size determined relative to measurement of lowest core count.

4. Summary

We have described the development of a parallel solver for multi-scale parquet quantum modeling of highly correlated materials. The code uses Newton's method with line search to solve the large system of affine quadratic equations. The large sparse Jacobian is too large to store and the action of matrix-vector multiply is computed analytically. Performance results on the Cray XT4/5 cluster suggest hybrid OpenMP and MPI programming technique can effectively use large numbers of cores to solve problems with several billion degrees of freedom.

Future development will focus on more sophisticated continuation method for generating good initial guesses and the exploration of effective preconditioners and iterative methods for solving the Jacobian system.

References

[1] A. BOUARICHA AND R.B. SCHNABEL, *Algorithm 768: TENSOLVE: A software package for solving systems of nonlinear equations and nonlinear least-squares problems using tensor methods*, ACM Trans. Math. Softw., 23 (1977), pp. 174–195.

[2] _____, *Tensor methods for large sparse systems of nonlinear equations*, Math. Program., 82 (1998), pp. 377–400.

[3] S. COHEN AND C. TOMASI, *Systems of bilinear equations*, Tech. Rep. CS-TR-97-1588, Computer Science Department, Stanford University, Stanford, CA 94305, 1997.

[4] N.J. HIGHAM AND H.-M. KIM, *Numerical analysis of a quadratic matrix equation*, IMA Journal of Numerical Analysis, 20 (2000), pp. 499–519.

[5] N.J. HIGHAM AND H.-M. KIM, *Solving a quadratic matrix equation by Newton's method with exact line searches*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 303–316.

[6] Y.V. MAKAROV, D.J. HILL, AND I.A. HISKENS, *Properties of quadratic equations and their application power system analysis*, Electrical Power and Energy Systems, 22 (2000), pp. 313–323.

[7] R.B. SCHNABEL AND P.D. FRANK, *Tensor methods for nonlinear equations*, SIAM J. Numer. Anal., 21 (1984), pp. 815–843.

[8] G.L.G. SLEIJPEN AND D. FOKKEMA, *BiCGstab(L) for linear equations involving nonsymmetric matrices with complex spectrum*, Electronic Transactions on Numerical Analysis, 1 (1993), pp. 11–32.

[9] C. SLEZAK, M. JARRELL, T. MAIER, AND J. DEISZ, *Multi-scale extensions to quantum cluster methods for strongly correlated electron systems*, Journal of Physics: Condensed Matter, 21 (2006). doi:10.1088/0953-8984/21/43/435604.

Parallel mesh control tools for unstructured meshes

Min Zhou,[1] Aleksandr Ovcharenko,[1] Ting Xie,[1] Seegyoung Seol,[1] Kenneth E. Jansen,[2] Onkar Sahni,[1] and Mark S. Shephard[1]*

[1]Rensselaer Polytechnic Inst, Sci Computat Ctr, Troy, NY12180
[2]University of Colorado at Boulder, Boulder, CO80309

E-mail: *shephard@scorec.rpi.edu

Abstract. The ITAPS center is developing services for unstructured mesh operations that operate on massively parallel computers including base services to effectively support evolving mesh partitions that take advantage of local and/or neighborhood information to eliminate syncronization steps and reduce communications. This paper overviews base services for controlling mesh partitions and controlling messages that are communicated within processor neighborhoods.

1. Introduction

Unstructured adaptive mesh methods support the effective and reliable analysis of complex physical behaviors modeled by partial differential equations over general three-dimensional domains. Although well defined adaptive meshes can have two to three orders of magnitude fewer elements than a more uniform mesh for the same level of accuracy, there are many complex simulations where the meshes required are so large that they can only be solved on massively parallel computers. The execution of a simulation on a massively parallel computer requires the mesh be distributed over the computing nodes and cores of the parallel computer through a partitioning of the mesh in which the amount of computation required for each part is nearly equal and the amount of inter-processor communication between parts is as small as possible. Given the fact that the meshes needed will not fit within the memory of even a single node, the methods developed to manage the mesh in parallel must address: (i) Understanding how the mesh entities and their adjacencies relate to computation and/or memory use. (ii) Performing multiple dynamic load balancing steps to account for the changing load balance caused by operations as adaptive mesh modification. (iii) Constructing and managing the mesh partition must be performed using effective scalable methods to avoid that step becoming more dominant as we move toward exascale machines.

A key conclusion the above factors point to is that parallel adaptive simulation on massively parallel computers requires all steps be efficiently executed in parallel on a partitioned mesh and that the mesh be effectively repartitioned as needed at various steps in the simulation.

Although a reasonable set of procedures and methods had been developed for parallel mesh adaptation when a few hundred processors were used [1,2], recent efforts to move to massively parallel computers with 100,000's of compute cores [3,4,5] indicated the need to develop a number of additional capabilities [6]. This paper focuses on two of these capabilities. The first capability is an extension of the partition model to support multiple parts per processor. Although we have run into multiple scenarios where this capability is useful, the overriding one in massively parallel adaptive simulations is when the total size of the mesh grows by one or two orders of magnitude during a

simulation, yielding a situation where it is desirable to increase the number of cores being used in subsequent steps. The second capability is a predictive load balancing procedure that is used to drive a repartitioning of the mesh before a mesh adaptation step with the goal of having nearly a balanced mesh partition after the mesh is adapted. Although this capability is potentially useful in maintaining computational load balance during mesh adaptation, its primary goal is to control the amount of memory used by each part so that memory usage is nearly equal with no parts requiring substantially more than the average memory usage. On some of the most effective massively parallel computers, exceeding the memory on a single core will lead to failure or termination of the calculation.

2. Partition Control on Massively Parallel Computers

Graph/hypergraph-based algorithms are most effective for partitioning unstructured meshes [7,8,9]. In our work on parallel computations at extreme scale, specific issues were observed [6]. One issue relates to the cost and reliability of applying parallel dynamic load balance over all cores using a graph of the entire distributed mesh, which is referred to as global partitioning. As core counts increase, the quality of global partitions can degrade. In addition there is a reliability issue that arises in some current graph/hypergraph-based partitioner implementations that, in our experience, often fail for partitions requiring large number of parts (≥100K). The second issue is related to the characteristics of implicit parallel mesh-based analysis which has the two primary work components of equation formation and equation solution which require load balance of different types of mesh entities (elements and vertices in the present case). Graph/hypergraph-based partitioning uses one type of mesh entities as graph nodes, thus, the balance of other mesh entities may not be optimal. Partition improvement methods have been developed to account for a balance of multiple entity types within a single partition. Those issues are addressed in our work by the combined strategy of global and local graph partitioning plus iterative improvement using the algorithm developed in [6].

Two partitioning categories are defined based on data provided to graph/hypergraph-based partitioner: *global partitioning* which considers both intra-processor and inter-processor graph edges, and *local partitioning* which considers only intra-processor graph edges. Global partitioning considers the complete set of graph edges and provides a balanced partition with well controlled inter-part communication. Local partitioning considers only the on-processor (intra-processor) graph edges and nodes, without knowing the existence of graph nodes and edges on other processors. In this case, partitioning is carried out independently on each processor, as a serial process, which can be executed in an embarrassingly parallel fashion on all processors. At large core counts local partitioning requires much smaller compute time where a global partitioning implementation may fail. However, as local partitioning is repeated, the quality of the partition will decrease due to the compounding of imbalance in each step. Local partitioning is not optimal, but provides good starting partitions that can easily be improved by iterative algorithms.

When very large core counts are considered, the problems observed in the partition obtained from the graph/hypergraph-based procedures are limited to a number of heavily loaded parts (based on mesh vertices), referred to as spikes. Thus, scalability of the equation solution phase is limited by these spikes. The diffusive approach, Local Iterative Inter-Part Boundary Modification (LIIPBMod) developed in [6], reduces spikes by migrating selected mesh entities from relatively heavily loaded parts to less loaded neighboring parts. On heavily loaded parts, the mesh vertices on the inter-part boundary are traversed and the ones bounding a small number of elements are identified. Then the elements adjacent to selected vertices are migrated to the lightly loaded neighboring part. Figure 1 explains the algorithm using three, 2D examples for clarity. The procedure has been implemented for 3D meshes. By this local inter-part boundary adjustment, the vertex imbalance is improved while only modestly perturbing the element balance. The procedure is repeated for several iterations to achieve the desired improvement to the vertex balance. In the extreme scale simulations, the global and local partitioning are used in a combined manner. i.e., in the first step, the mesh is balanced globally into an intermediate number of parts, m, and in the second step, each of these m parts split independently to

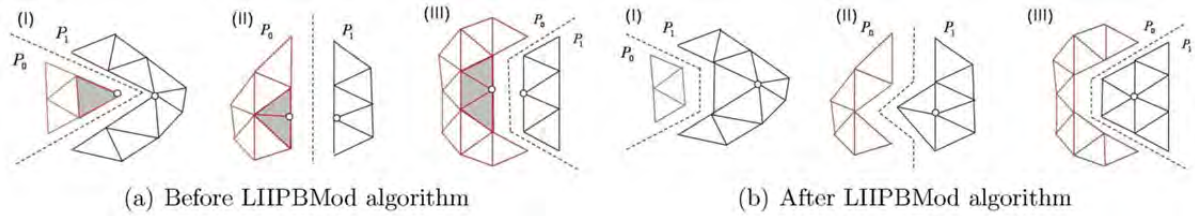


Figure 1. Partition before and after LIIPBMod.

n parts, which gives $m \times n$ parts in total. In the final step, the balance of the partition is improved by an iterative mesh entity migration procedure. This combination is much faster and more efficient compared to global partitioning.

An abdominal aorta aneurysm (AAA) model (see Figure 2) is used to study the partition improvement algorithm by using it in conjunction with graph/hypergraph-based procedures, namely ParMETIS PartKWay [9] and Zoltan Parallel HyperGraph partitioner PHG [7].

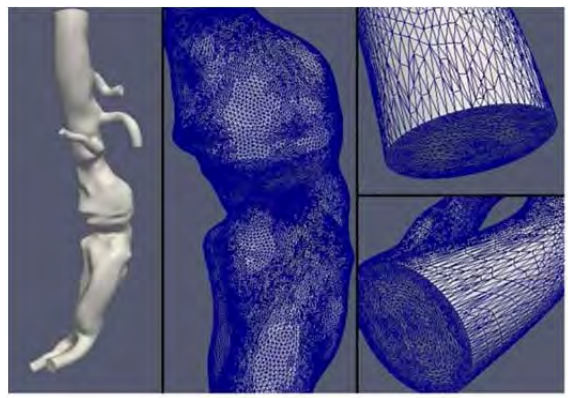


Figure 2. Geometry and mesh of an AAA model.

First consider a 1.07 billion anisotropic, tetrahedral element mesh created by applying mesh adaptation [10] on an AAA model. The global (ParMETIS PartKWay) and local (PHG) partitioning are combined to obtain different partitions with number of parts ranging from 4,096 to 294,912 [11]. The element imbalance is within 6% for all the partitions with up to 294,912 parts. However, the vertex imbalance is getting worse when the mesh is distributed to more and more parts. Each part of a globally balanced (1.025% element imbalance) partition with 4,096 parts splitting to 72 parts gives a partition with 294,912 parts in total. The element imbalance is 5.6%, but the vertex imbalance is 43.7%, which indicates 43.7% more work to do on the part with the highest number of vertices during the equation solution phase of the FEA. The LIIPBMod algorithm reduces the vertex imbalance dramatically to 17% while increasing the element imbalance to 15%. Table 1

Table 1. Strong scaling results of FEA on an AAA model with 1.07*B* elements up to 294,912 cores on JUGENE with and without LIIPBMod algorithm. *LMod* denotes LIIPBMod

1.07*B* element mesh num. of cores	Eqn. form.		Eqn. soln.		Total	
	Time	s_factor	Time	s_factor	Time	s_factor
4,096 (base)	390.17	1	455.51	1	845.68	1
294, 912	5.54	0.98	10.38	0.61	15.92	0.74
294, 912 (*LMod*)	5.82	0.93	9.14	0.69	14.96	0.79

presents the time usage of the FEA along with the scaling factors. It also compares the cases with and without using LIIPBMod algorithm. The equation solution phase is accelerated by 12% (from 10.38 to 9.14 seconds) due to better vertex balance, while the equation formation is slowed down a little from 5.54 to 5.82 seconds. The total time usage of the FEA is reduced from 15.92 to 14.96 seconds and the scaling factor is increased from 0.74 to 0.79. The time spent on the FEA is reduced by 0.96 second per 20 time steps, which means 78.6 cpu hours. In the real application, we usually run thousands of time steps per cardiac cycle, e.g., 5000. By using LIIPBMod algorithm, we save $78.6 \times 5000 = 393$

thousand cpu hours per cardiac cycle in analysis. The time spent on LIIPBMod algorithm is much smaller than the classical graph/hypergraph-based partitioner, which is negligible compared to solver [6]. An 8.56 billion element mesh was also run and yielded similar results.

3. Neighborhood Aware Message Packing

The Inter-Processor Communication Manager (IPComMan) is a package to reduce data exchange costs by exploiting communications of a local neighborhood for each processor. The neighborhood is the subset of processors exchanging messages with each other during a communication round, which in a number of applications is bounded by a constant, typically under 40, independent of the total number of processors. The library takes care of the message flow with a subset of MPI functions. The library provides several useful features (i) automatic message packing, (ii) management of sends and receives with non-blocking MPI functions, (iii) asynchronous behavior unless the other is specified, and (iv) support of dynamically changing neighborhood during communication steps. IPComMan takes care of memory allocation for both sending and receiving buffers, and manages the ones that can be reused without additional allocation. The library stores messages going to the same processor in contiguous memory. Thus, when sending or receiving a package, no additional memory copying is needed. The user may specify whether the size of each message is constant or arbitrary during a specific communication step. The fixed message size is taken by the library and used while extracting the messages. The arbitrary message size is put together with every message to correctly unpack the message upon arrival.

While initializing a communication library object, each processor specifies the neighbors it is going to communicate with. From that point, the library is concentrated on delivering messages between neighbors only, not touching other processors of the domain, when required. There are no collective calls during each communication round that remains within a neighborhood. There could be situations when it is not possible to a priori define all the neighbors for processors, i.e., new neighbors may be encountered during a communication step. For example, mesh modification operations may alter the neighborhood of specific processors. Consider the communication pattern presented in Figure 3. Processor P0 has in its neighborhood processors P1 and P2. P3 has P2 as the only neighbor, but after it has sent all the messages to P2 it finds out that there are some messages to be sent to P0. P3 includes P0 in its list of neighbours and begins to send packages to it. An increment of time before, say P0 finished sending to and receiving all the messages from P1 and P2, extracted them and proceeded to the next communication step. In that case packages from P3 to P0 would be lost, which will result in incorrect program behavior. To avoid this problem as new neighbors needing to communicate are encountered, one collective call at the end of communication round is performed to verify whether the global number of sends and receives match. When new neighbors are formed, the library continues to receive the messages identifying the new neighbors and communications. Note that at that time all the packages from existent neighbors are already received.

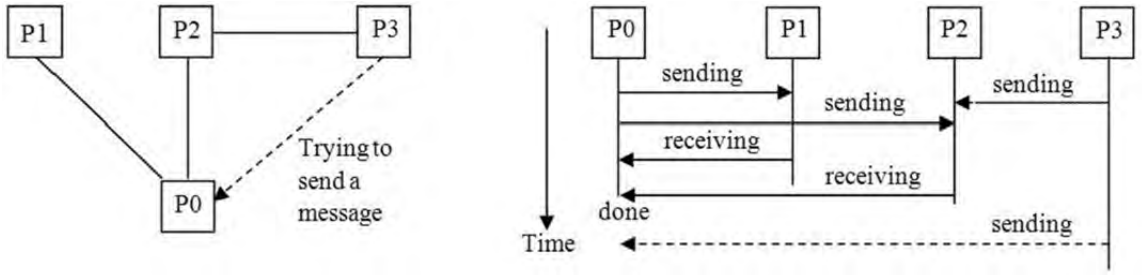


Figure 3. Communication paradigm with the neighborhood being changed.

Dynamic and irregular computation often results in an unpredictable number of messages communicated among the processors. Using IPComMan, there is no need to verify and send the

number of packages to be received at the end of sending phase. The last message sent from the processor to a neighbor contains the number of buffers expected to be received by the neighbor.

To measure the performance of mesh migration, an essential part of parallel adaptive unstructured mesh procedures [12], IPComMan was compared with the Autopack communication utility. Initial results on IBM Blue Gene/L [3] show that IPComMans ability to localize communication to neighborhoods to be independent of the total number of processors and its use of non-blocking functions allows it to continuously reduce the communication time as the number of processors increases. Even though the differences in communication times between Autopack [13] and IPComMan are not substantial at 128 processors, IPComMan is from 3 to 7 times faster in the 4096 processor case.

Although the total communication time for IPComMan is reduced, the scaling for the process does fall with increased processor count. It is important to note though that the mesh adaptation process that involves mesh migration procedures is not well balanced in terms of the communication load per part. However, additional efforts on the scalability are desired. This includes several aspects of managing sends and receives during the communication phase while interacting with the computation part, and keeping efficient use of the buffer memory. The options are being considered to eliminate all the collective calls with the neighborhood concept in IPComMan, although additional analysis is required to see if these approaches lead to the reduction of the communication time in oppose of using collective call with the processor count growth.

4. Closing Remark

As the number of cores used in our parallel adaptive simulations continue to increase it is becoming increasingly important to ensure that the tools used to adapt the mesh and load balance it for the next set of solution steps are efficient and scale, while at the same time produce a partitioning of the mesh that ensures the solver will scale well (preferably strong scaling). This paper has outlined two services that have been recently to help meet these needs.

References

[1] Flaherty J, Loy R, Özturan C, Shephard M, Szymanski B, Teresco J, and Ziantz L. 1996. *Applied Numerical Mathematics*, **26** 241–263.
[2] Shephard M, Flaherty J, Bottasso C, de Cougny H, Ozturan C, and Simone M. 1997. *Parallel Computing*, **23** 1327–1347.
[3] Sahni O, Zhou M, Shephard M, and Jansen K. 2009. *Proceedings of the SC09* (Springer, Berlin).
[4] Shephard M, Jansen K, Sahni O, and Diachin L. 2007. *Journal of Physics: Conference Series* **78**-012053 012053.
[5] Zhou M, Sahni O, Kim H, Figueroa C, Taylor C, Shephard M, and Jansen K. 2010. *Computational Mechanics*, **46**(1) 71–82.
[6] Zhou M, Sahni O, Shephard M, Devine K, and Jansen K. 2010. *SIAM J. Sci. Comp.*, 2010, (under review).
[7] Boman E, Devine K, Fisk L, Heaphy R, Hendrickson B, Leung V, Vaughan C, Catalyurek U, Bozdag D, and Mitchell W. 1999. Zoltan home page http://www.cs.sandia.gov/Zoltan.
[8] Hendrickson B and Leland R. 1995. *Proc. Supercomputing '95* (ACM).
[9] Karypis G and Kumar V. 1996. *10th Intl. Parallel Processing Symposium*, 314–319.
[10] Sahni O, Müller Y, Jansen K, Shephard M, and Taylor C. 2006. *Comp. Meth. Appl. Mech. Engng.*, **195** 5634–5655.
[11] Zhou M, Sahni O, Xie T, Shephard M, and Jansen K. 2010. *Journal of Supercomputing*, Under review.
[12] Li X, Shaphard M, and Beall M. 2005. *Comp. Meth. Appl. Mech. Engng.*, **194** 4915–4950.
[13] Loy R Autopack user manual science Division Argonne National Laboratory, http://wwwunix.mcs.anl.gov/autopack/.

Verification and validation of current models of thermonuclear-powered supernovae using large-scale simulations

Ray Bair,[1] Joe Bernstein,[2,3] Chris Daley,[4] Benedikt Diemer,[4] Anshu Dubey,[4] Robert Fisher,[5] Jonathan Gallagher,[4] Carlo Graziani,[4] Randy Hudson,[4,6] George Jordan,[4] Rick Kessler,[4] Yeunjin Kim,[4] Shravan Kumar,[4] Don Lamb,[4] Dongwook Lee,[4] John Norris,[4,6] Mike Papka,[4,6] Paul Rich,[4] Katherine Riley,[3] Daan van Rossum,[4] Dean Townsley,[7] Klaus Weide,[4] and Guohua Xia[4]

[1]CELS Directorate, Argonne National Laboratory
[2]HEP Division, Argonne National Laboratory
[3]ALCF, Argonne National Laboratory
[4]DOE NNSA/ASC Alliance Astrophysical Center for Thermonuclear Flashes, Department of Astronomy & Astrophysics, University of Chicago
[5]Department of Physics, University of Massachusetts, Dartmouth
[6]MCS Division, Argonne National Laboratory
[7]Department of Physics and Astronomy, University of Alabama

Abstract. We report initial results of a verification study to determine the fundamental properties of buoyancy-driven turbulent nuclear combustion and a comprehensive, systematic program to validate current models of Type Ia supernovae.

1. Introduction

Observations using Type Ia (thermonuclear-powered) supernovae (SNe Ia) led to the discovery of dark energy and provide one of the most promising methods for determining its properties. Most scientists believe that using these explosions to accomplish the latter will require a much better understanding of them. Two major challenges face numerical astrophysicists in the Type Ia supernova (SN Ia) field: (1) several key physical processes in these explosions are not fully understood, including buoyancy-driven turbulent nuclear combustion; and (2) very few simulations of the current models of SNe Ia have been done, making it difficult to determine which of these models is favored by observations, and even more, what values of the many parameters specifying these models are consistent with observations. We report the results of extensive large-scale, 3D verification simulations of buoyancy-driven turbulent nuclear combustion for planar flames in a channel, flame bubbles in an open domain, and flame bubbles in a white dwarf star. We have also begun a comprehensive, systematic validation of current models of SNe Ia, using through large-scale, whole-star 3D hydrodynamic simulations of these explosions using FLASH [2] and radiation transfer calculations using Sedona [5] to compare the light curves and spectra predicted by these models with high-quality data obtained by the SDSS-II Supernova Survey team and its collaborators [6]. We describe the initial results of these efforts below.

2. Buoyancy-Driven Turbulent Nuclear Combustion

We are attempting to provide definitive answers to three questions regarding buoyancy-driven turbulent nuclear combustion:

- Is it possible to describe the burning rate of a turbulent flame by a single characteristic turbulent timescale? If so, what scale dominates the flow?
- Under what conditions does a flame transition from the flamelet regime to the distributed burning regime?
- How does buoyancy-driven turbulent nuclear combustion in a stratified medium differ from turbulent nuclear combustion in a homogeneous isotropic turbulent background?

To answer these questions, we have conducted large-scale 3D simulations of (i) planar flames in a rectilinear computational domain in which g and ρ are constant; (ii) flame bubbles in an open domain, which we simulate using a rectilinear computational domain having large lateral dimensions, in which g is constant and ρ is decreasing; and (iii) flame bubbles in a white dwarf star, in which g is increasing and ρ is decreasing. Figure 1 shows a simulation of the first physical situation; Figures 2 and 3 show simulations of the latter two physical situations. We have made the following discoveries about buoyancy-driven turbulent nuclear burning:

- The rate of nuclear burning appears to be governed by the length scale corresponding to the flame polishing length λc [7] (see Figure 4).
- "Self regulation" is a physical process in which changes in the area of the flame exactly compensate for changes in the laminar flame speed, causing the nuclear burning rate to be independent of the laminar flame speed. Our simulations suggest self regulation is the result of the fundamental properties of buoyancy-driven turbulent nuclear combustion, and so occurs not only for planar flames in closed computational domains—where it has been observed previously—but also for flame bubbles in open domains, which is directly relevant to the burning that occurs during the deflagration stage of SNe Ia.

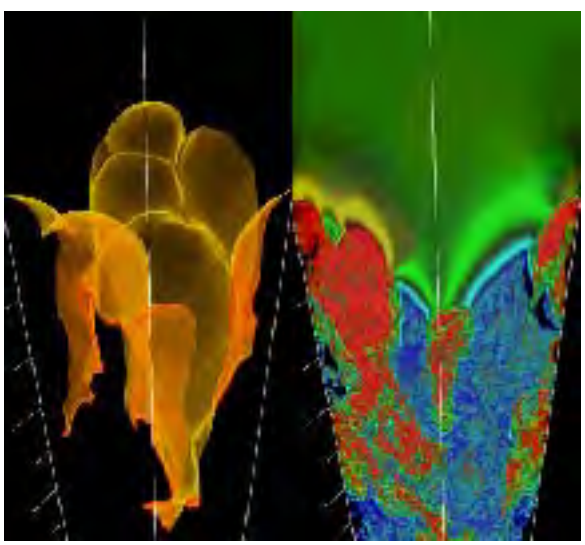


Figure 1. Frames showing two different ways of visualizing the physical properties of a simulation of buoyancy-driven turbulent reactive flow at a moment in time for a flame speed s = 30 km s^{-1}. Left frame: volume rendering of the flame surface; right frame: volume rendering of the velocity field generated by the flame [10].

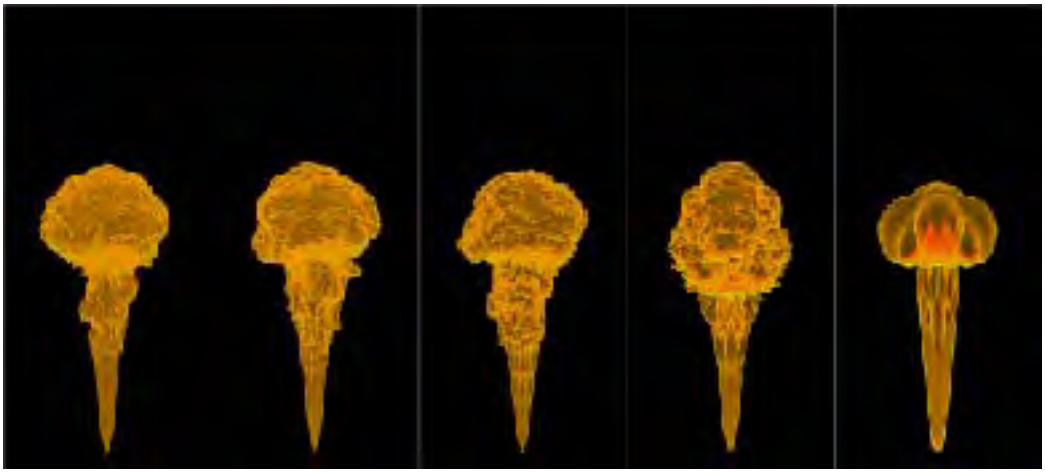


Figure 2. Resolution study of simulations of a flame bubble in an open domain (mocked up by a rectilinear domain having large lateral dimensions) in which g is constant and ρ is decreasing.

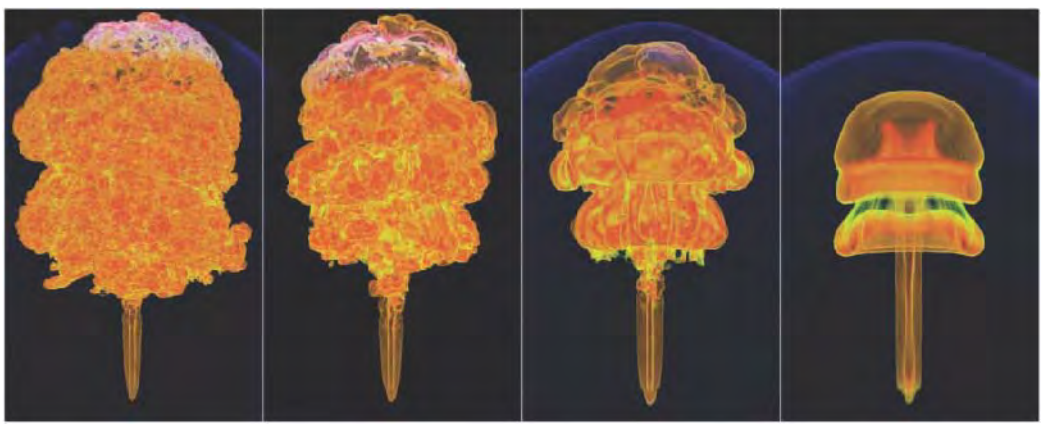


Figure 3. Resolution study of simulations of a flame bubble in a white dwarf star in which g is increasing and ρ is decreasing.

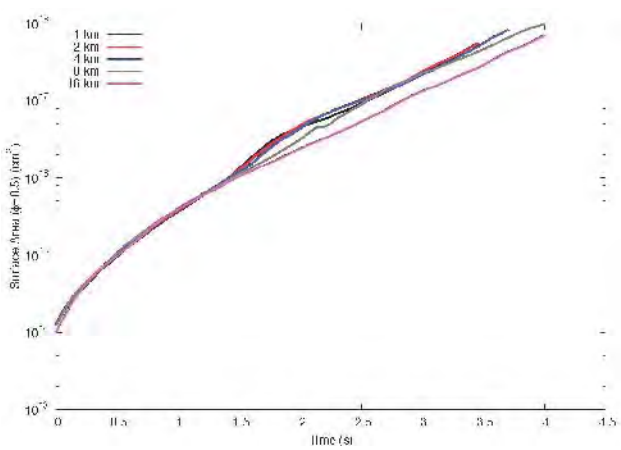


Figure 4. Flame area as a function of time for different spatial resolutions for flame bubble simulations in an open domain (mocked up by a rectilinear domain having large lateral dimensions) with constant acceleration of gravity g and decreasing density ρ. The results demonstrate convergence with resolution for spatial resolutions of 8 km or better; i.e., simulations that resolve the flame polishing length λc.

- The condition Ka = 1, where Ka is the so-called Karlovitz number, has been used in the SN Ia field as the criterion for when a flame transitions from the flamelet regime to the distributed burning regime (see, e.g., [1]). Our simulations show the flame remains in the flamelet regime even for Ka » 1. This is important because transition to the distributed burning regime is a pre-requisite for initiation of a detonation in the deflagration-to-detonation model.

3. Validation of Current Type Ia Supernovae Models

We have extended our large-scale, whole-star 3D hydrodynamic simulations of SN Ia models using FLASH [2] from the GCD models [9,4,3,8] (see Figure 5) to the pure deflagration and deflagration-to-detonation models. Taking the data from these simulations as input for radiation transfer calculations using Sedona [5], we find the following:

- The light curves predicted by the GCD model can be fit reasonably well by the MLCS2k2 and SALT2 data-driven models the supernova community uses to fit to the light curves of individual SNe Ia [6], showing that the predicted light curves are similar to those observed.
- Preliminary results based on 2D simulations show the light curves predicted by the GCD model are consistent with the Phillips relation between peak B-band magnitude and stretch and the inference from observations that most observed SNe Ia produce 0.5-1 solar masses of nickel (see Figure 6).
- Viewing asymmetric supernovae from different directions may contribute significantly to the anomalous scatter in the calibration of SNe Ia as “standard candles” (see Figure 6).

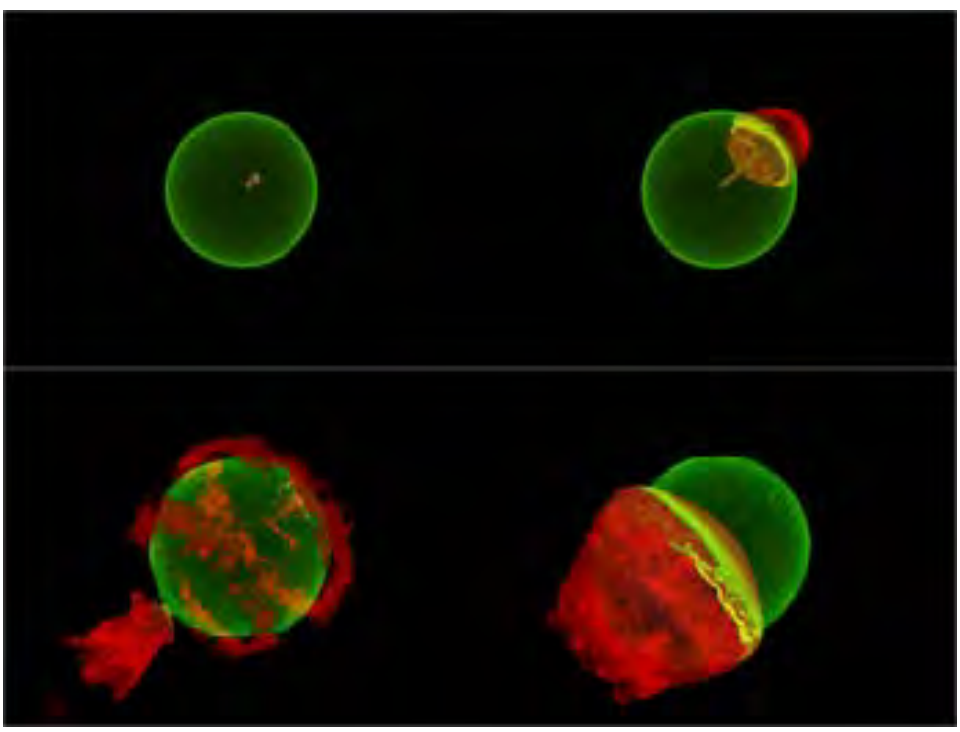


Figure 5. Images showing extremely hot matter (ash or unburned fuel) and the surface of the star at different times for an 8-km resolution simulation of the GCD model starting from initial conditions in which an 16-km radius hot bubble is offset 80 km from the center of the star. Times are (a) 0.5 s, soon after the bubble becomes Rayleigh-Taylor unstable and develops into a mushroom shape, (b) 1.0 s, as the bubble breaks through the surface of the star, and (c) 2.03 s, when the hot ash has flowed over the surface of the star and has begun to collide, and (d) 2.23 s, as the detonation wave sweeps through the star [9,4,3,8].

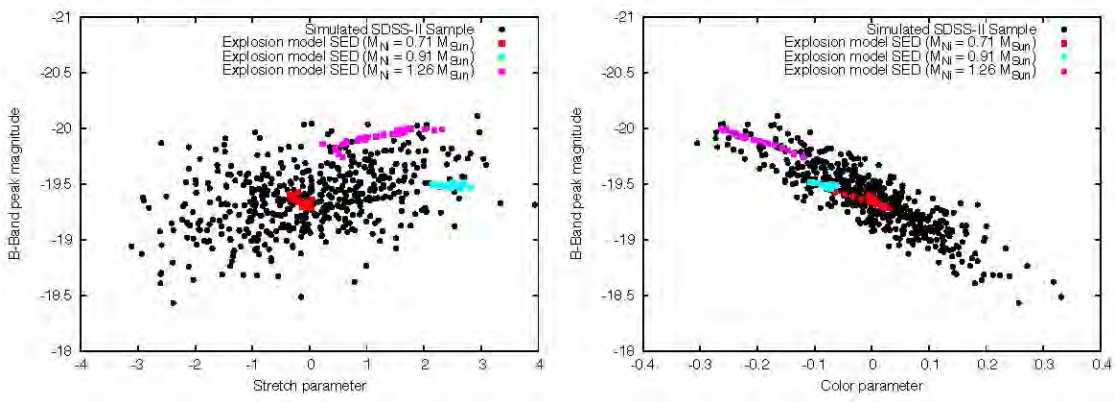


Figure 6. Left panel: Points corresponding to 30 different viewing angles predicted by simulations of the GCD model of SNe Ia for three different initial conditions that produce 0.71, 0.91, and 1.26 solar masses of nickel, superimposed on a scatter plot of a simulated SDSS-II sample of SNe Ia in the (stretch, B-band peak magnitude)-plane, where stretch is a measure of the rate an SN Ia fades and B-band peak magnitude is a measure of the peak luminosity of a SN Ia. Right panel: Same, except in the (color, B-band peak magnitude)-plane. These preliminary results, which are based on 2D simulations, show the light curves predicted by the GCD model are consistent with the Phillips relation between peak B-band magnitude and stretch and the inference from observations that most observed SNe Ia produce 0.5–1 solar masses of nickel.

References

[1] Aspden A J, Bell J B, Day M S, Woosley, S E, and Zingale M. 2009. *ApJ* **689** 1173–1185.
[2] Fryxell B, et al. 2000. *ApJSuppl* **131** 273–334.
[3] Jordan G C IV, et al. 2009. In Pogorelov NV, et al., eds., *Astronomical Society of the Pacific Conference Series* **406** 92.
[4] Jordan G C IV, et al. 2008. *ApJ* **681** 1448–1457.
[5] Kasen D, Thomas R C, and Nugent P. 2006. *ApJ* **651** 366–380.
[6] Kessler R, et al. 2009. *ApJSuppl* **185** 32–84.
[7] Khokhlov A M. 1995. *ApJ* **449** 695–713.
[8] Meakin C A, et al. 2009. *ApJ* **693** 1188–1208.
[9] Townsley D M, et al., 2007. *ApJ* **668** 1118–1131.
[10] Townsley D M, et al. 2008. In *Journal of Physics: Conference Series* **125** 012009 IOP Publishing

Extreme-Speed Scalable Direct Sparse Solvers for Heterogeneous Supercomputing—An Enhancement to the LAPACKrc Library

Juan Gonzalez[1] and Rafael C. Núñez

[1]Accelogic, 1830 Main Street, Suite 204, Weston, FL, U.S.A.

E-mail: juan.gonzalez@accelogic.com

Abstract. At the core of DOE's scientific priorities is a broad need for more computing power and supercomputing infrastructure. An estimated 70% of computing cycles spent globally in the HPC ecosystem are used to solve large-scale linear algebra computational problems. Linear algebra is at the core and constitutes the primary bottleneck of important DOE research problems in fusion energy, nuclear accelerator modelling, circuit simulation, and weather modelling, among other research challenges. As part of an ambitious research program co-funded by DOE, NASA, and the Department of Defense, Accelogic is spearheading the development of LAPACKrc, a groundbreaking family of FPGA-based linear algebra solvers able to achieve speedups larger than 100x with a single chip ("rc" stands for "reconfigurable computing," a radically new computing paradigm that is changing the way high performance computing is done today). Recent efforts on the LAPACKrc research program have focused on producing FPGA-based direct sparse solvers—a key functionality still missing in the current LAPACKrc solver suite. Our latest direct sparse solver prototype demonstrates a speedup of up to 125x (compared against state-of-the-art CPU direct sparse solvers), which provides support for broad-based science/engineering breakthroughs.

1. Introduction

Improvements in High-Performance Computing (HPC) have become one of the primary means of addressing the many critical needs the scientific community continuously faces. These improvements are more needed every day, as HPC supports the discipline of "modeling and simulation," now considered to be a fundamental part of the scientific method, replacing experimentation when the latter becomes prohibitive in terms of cost and/or difficulty. Energy, climate modeling, oil & gas exploration, health, and astrophysics, are among the numerous scientific areas always avid for increased computational power.

One approach to boosting computational capacity in large-scale codes while reducing costs is using heterogeneous architectures —*i.e.,* programming environments that involve different types of processor architectures operating simultaneously (*e.g.,* CPUs, FPGAs, GPUs, Cell Broadband engines). The authors in [1] consider "the emergence of multicore/manycore and heterogeneous architectures the single biggest change in scalable computing in the past decade, with <u>effective use of these new node architectures</u> critical to reducing the performance gap between the peak performance of the hardware and the realized performance of the applications."

Exploiting heterogeneous architectures for scientific applications is becoming mainstream. Supercomputing technologies such as Cray's XT5h [2] and SGI's RASC [3] have been in the market

for several years now, and it is common to find commercial supercomputer components that allow the seamless addition of FPGA or GPU computing nodes to an existing supercomputing network - transparently. On the software side, powerful libraries and programming tools are either available or at an advanced development stage for von-Neumann heterogeneous systems (*e.g.,* multicore, GPUs, etc), as it is evidenced by the CUBLAS [4], PLASMA, and MAGMA [5] libraries.

For heterogeneous non-von-Neumann architectures (*e.g.,* systems enriched with FPGA accelerators), the development of software and numerical libraries is more challenging – yet potentially much more rewarding – and has been happening at a slower pace. FPGAs are specialized ―programmable logic" devices able to execute specific routines several orders of magnitude faster than a traditional von Neumann CPU – while maintaining their entire (software based) programmability. In addition to quantum increases in speed, FPGA-systems are also ―green," rendering between ten and hundreds of times more energy-efficiency than conventional CPUs.

FPGA computing technology provides a flexible framework that allows tackling **bottlenecks** not efficiently handled by von Neumann solutions. For example, simultaneously adapting arithmetic units for each of the bottlenecks in the computational chemistry code NWChem [5], is a task that can be executed better by flexible/adaptable supercomputing architectures. One of the methods used in NWChem is the Hartree-Fock (HF) method. The computational speed in this *ab initio* electronic structure algorithm is conditioned by two core functions, namely, Fock matrix formation and eigenproblem solution. The main bottleneck of the algorithm can be either the Fock matrix formation or the eigenproblem solution. When the code is executed on a small number of processors, matrix formation is the primary bottleneck, whereas, when the code is executed on thousands of processors, matrix formation is efficiently scaled thus shifting the primary bottleneck to the eigensolver (which is scalable up to about 300 processors today.) These two steps are inherently sequential, creating an opportunity for an FPGA-enhanced accelerated solution. In FPGA computing, instead of having a single computing architecture that scales very well and provides computing performance for only one of the two bottlenecks, we can have two different architectures alternating operations in the same FPGA chip. These two architectures can be optimized for the solution of each particular problem. In a different flavor of FPGA enhancement, it is possible to connect a small group of FPGA computing units into a traditional CPU-based supercomputing network, so that we take advantage of the scalability of the matrix formation in thousands of CPUs (currently achieved by NWChem), and of accelerated eigensolvers in a small number of FPGAs (a solution that is possible through the use of specialized FPGA numerical libraries), to achieve groundbreaking end-to-end acceleration. This is a perfect example of a real heterogeneous supercomputing architecture in action, enabled through adaptive computing software.

Aimed at providing the needed software substrate for the effective exploitation of FPGA accelerators in current supercomputers, Accelogic is developing LAPACKrc, a groundbreaking family of linear algebra solvers that increase computational speed 100+ times using hybrid CPU/FPGA computing [13], [14], [15]. In the remainder of this paper, we introduce recent advances in the development of this library, particularly focused on incorporating FPGA-accelerated direct sparse solvers into LAPACKrc.

2. The challenges of heterogeneous CPU/FPGA supercomputing

In order to achieve consolidation as a viable alternative that provides very-low-cost and energy-efficient HPC—at both the supercomputer and the workstation levels, FPGA computing has faced three fundamental challenges:

(a) Making available FPGA hardware platforms well-suited to scientific computing;
(b) Making available advanced programming tools and compilers for FPGA computing design, comparable to those currently used in traditional von Neumann design; and,
(c) Providing software and numerical libraries that are fast, accurate, portable, and easy to use, thus allowing for widespread adoption of FPGA acceleration by the numerical computing community, and by domain scientists.

Fortunately, with the evolution of FPGA computing during the last 10 years, most of these challenges are being successfully addressed. With respect to (a), supercomputer vendors already offer an impressive array of FPGA-based computer systems—most notably the SGI RASC [3], Cray's XT5h [1], the SRC-7 system [7], DRC Accelium Platform [8], and Convey HC-1 system [9]. As for (b), a significant number of programming tools and C-like compilers for FPGAs are already available both within the open-source (*e.g.,* SPARK [10]) and commercial (*e.g.,* Mitrion platform [11], Impulse C [12]) communities.

Most success in (c) has been achieved and is ubiquitous for integer-arithmetic applications such as encryption and bioinformatics. In the floating-point arena, LAPACKrc offers a powerful array of numerical libraries that exploit the presence of FPGAs in a supercomputer network, under a paradigm that provides maximal speed performance, but is yet portable and easy to use for the domain scientist.

3. Enabling heterogeneous supercomputing for scientific applications: The LAPACKrc numerical library

Sca/LAPACKrc (or simply LAPACKrc) is a family of FPGA-based linear algebra solvers able to achieve speedups larger than 100x per FPGA chip. The goals of the LAPACKrc program are to 1) develop the fundamental FPGA computing algorithmic technology for high-speed FPGA-enhanced dense and sparse linear algebra; 2) produce solvers and their underlying math components as portable and easy-to-use utilities; and 3) aggressively integrate the LAPACKrc technology into higher-level solutions for DOE and its other program sponsors. LAPACKrc incorporates the traditional linear algebra functionality of libraries like LAPACK, ScaLAPACK, PETSc, and MUMPS. The library is compatible with traditional CPU-based High-Performance Computing (HPC) hardware, and can be used to accelerate current HPC codes via plug-in enhancements with hybrid accelerators such as the SRC-7 or the Cray XT5h [13], [14], [15]. Figure 1 illustrates the core components of the LAPACKrc library. It is worth noting that, although initially focused on FPGA acceleration, LAPACKrc is designed to integrate and "adaptively" squeeze performance from all the components of a heterogeneous supercomputer (*e.g.,* CPUs, GPUs, FPGAs) when working together to solve a problem.

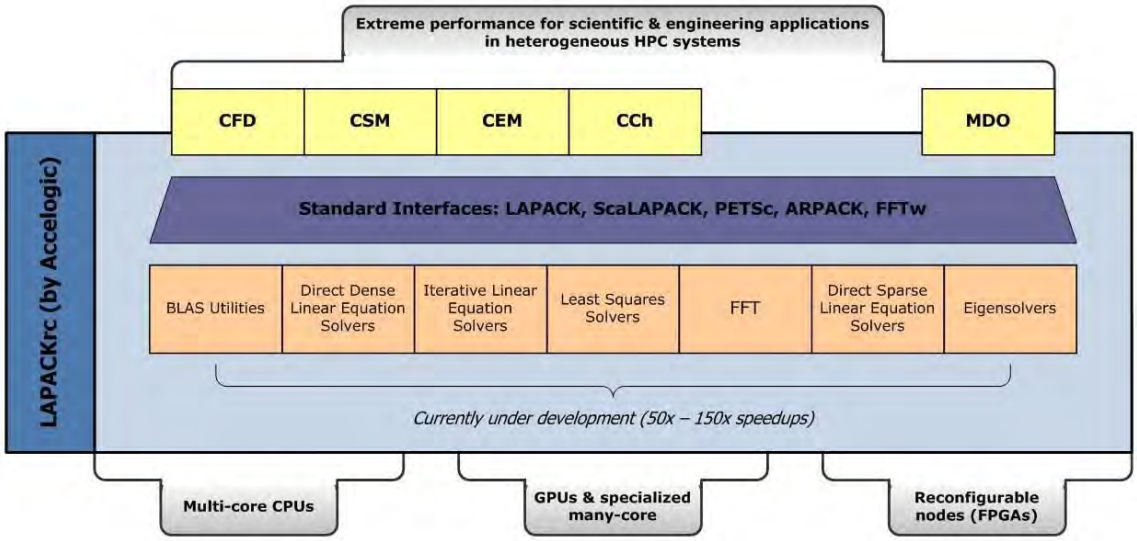


Figure 1. Core components of the LAPACKrc library. LAPACKrc prototypes have demonstrated speedup factors of up to 150x for a variety of large-scale sparse and dense linear algebra problems.

4. Reducing time to solution in direct sparse computations

One of our most recent efforts in enhancing LAPACKrc has focused on incorporating direct sparse solver functionality. Our design is a collaborative architecture, with selected operations of the sparse direct solver executed in the CPU, and other numerically intensive kernels executed in the FPGA. Our sparse Cholesky solver is based on CHOLMOD [16]. CHOLMOD is used by MATLAB's _A\b' operator when A is sparse symmetric positive definite. Our direct sparse computational engine is based on a many-core arithmetic unit (up to 1,640 cores per FPGA in our first prototype) for which operations are controlled by a dynamic scheduler.

We benchmarked our solver with all of the symmetric positive definite matrices in the University of Florida Matrix Collection [17], and attained speedups ranging from 3x for small matrices, up to 125x for large matrices. Figure 2 summarizes our results. We compared the performance of our solver against the fastest CPU solver (CHOLMOD [18]) running on the fastest CPU core (Intel Xeon Woodcrest [19]). The processing performance of our Cholesky solver (and of sparse Cholesky solvers in general) depends on the characteristics of the problem being solved. In general, the problem becomes more complex (*i.e.,* more time to solution) as the number of floating-point operations per nonzero entry in the factor L (of $A=LL^T$) increases. An important capability of this technology lies in its ability to provide more speedup as this ratio of flops per nonzeros increases.

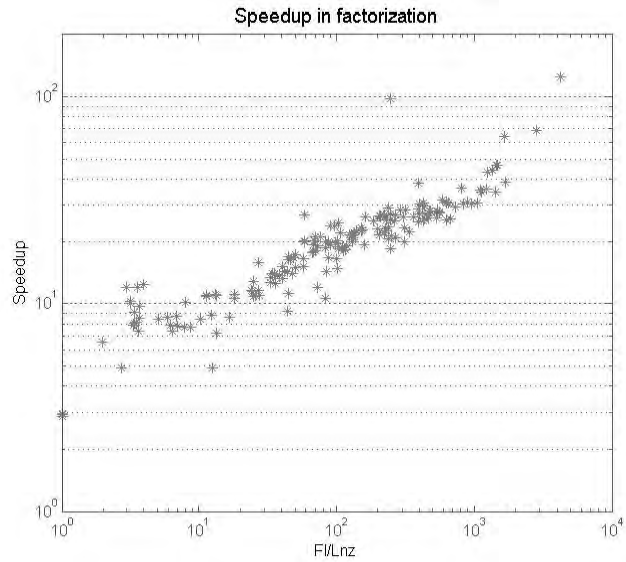


Figure 2. Speedup of LAPACKrc's sparse Cholesky factorization method, compared against CHOLMOD. The LAPACKrc solver runs on an Intel Xeon Woodcrest / Xilinx Virtex 6 system. CHOLMOD runs on an Intel Xeon Woodcrest @3GHz, on Ubuntu Linux.

Conclusions and future work

One approach to boosting computational capacity in large-scale codes while reducing costs is using heterogeneous architectures enhanced with accelerators such as FPGAs and GPUs. These types of systems have been in the market for several years now. Traditional supercomputers can be enhanced transparently by adding a number of accelerators able to boost performance —adaptively" for targeted numerical routines. The challenge in the proper exploitation of accelerator hardware is in well-designed software and numerical libraries that allow maximal speed and compatibility with the legacy CPU system, while maintaining portability and ease of use. Numerous efforts are starting to provide important software components, libraries, and development tools for accelerator-enhanced supercomputers. The development of FPGA-enhanced software and numerical libraries is more challenging than that of other accelerators – yet potentially much more rewarding in terms of overall speedup, cost, and power consumption. The LAPACKrc library represents the first industrial-quality family of linear algebra solvers intended for heterogeneous FPGA-enhanced HPC architectures, capable of speedups exceeding 100x per FPGA processor. The LAPACKrc library is currently integrating a large-scale sparse Cholesky factorization module that has reached up to 120x acceleration for symmetric/positive definite matrices in the sparse matrix collection. Future work includes, in addition to productization of the current prototype, the creation of additional methods for linear algebra.

Acknowledgments
This material is based upon work supported by the Department of Energy, U.S. Air Force, and NASA, under award numbers DE-SC-0001191, DE-FG02-07ER84728, DE-FG02-08ER85136, FA9550-09-C-0052, NNX07CA19P, and NNX10RA04C. We also wish to thank Tim Davis from the University of Florida and Jack Dongarra from the University of Tennessee for their consultation on this project.

Disclaimer
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References
[1] Geist A, Colella P, Heroux M 2008 Final Report *Workshop on CS/Math Institutes and High Risk/High Payoff Technologies for applications (2008)*
[2] Cray XT5h Supercomputer. http://www.cray.com/downloads/CrayXT5hBrochure.pdf (2008)
[3] SGI RASC Technology. http://www.sgi.com/products/rasc/ *(Sep 2006)*
[4] NVIDIA Corporation. CUBLAS Library Manual. Santa Clara, CA. *(Mar 2008)*
[5] Agullo E, et al 2009 Numerical linear algebra on emerging architectures: The PLASMA and MAGMA projects *Journal of Physics: Conference Series (Vol. 180, 2009)*
[6] Kendall R 2000 High performance computational chemistry: an overview of NWChem a distributed parallel application *(Computer Physics Communications 128: 260–283)*
[7] SRC-7 Overview. http://www.srccomputers.com/products/src7.asp. *(Feb 2008)*
[8] DRC Accelium. http://www.drccomputer.com/pdfs/DRC_Accelium_Overview.pdf *(2010)*
[9] The Convey HC-1: The World‘s First Hybrid-Core Computer. http://www.conveycomputer.com/Resources/HC-1%20Data%20Sheet.pdf *(2010)*
[10] Gupta R, Dutt N, Nicolau A 2004 SPARK: A Parallelizing Approach to the High-Level Synthesis of Digital Circuits. *Kluwer Academic Publishers*
[11] The Mitrion Platform. http://www.mitrionics.com, *(2010)*
[12] Impulse CoDeveloper C-to-FPGA Tools. http://www.impulsec.com/products_universal.htm.
[13] Veley D, Camberos J 2007 Computational Uncertainty in Reconfigurable Computing *NATO RTO Applied Vehicle Technology Panel (AVT) Symposium (Athens, Greece, Dec 2007)*
[14] Gonzalez J, Nunez R C 2008 LAPACKrc: A portable linear algebra library for extreme-speed acceleration of HPC codes using FPGAs *DoD High Performance Computing Modernization Program, 2008 Users Group Conference (Seattle, WA, July 14-18, 2008)*
[15] Gonzalez J, Nunez R C 2009 LAPACKrc: Fast linear algebra kernels/solvers for FPGA accelerators *Journal of Physics: Conference Series (Vol. 180, 2009)*
[16] Chen Y, et al 2009 Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate *ACM Trans. Math. Software (Vol 35, No. 3, 2009)*
[17] Davis T 1997 The University of Florida Sparse Matrix Collection, http://www.cise.ufl.edu/research/sparse/matrices *(NA Digest, vol. 97, no. 23, June 7, 1997)*
[18] Gould N, Hu Y, Scott J A. 2006 Complete results from a numerical evaluation of sparse direct solvers for the solution of large, sparse, symmetric liner systems of equations *NAG Internal Report 2005-1 (revision 2). Rutherford Appleton Laboratory (March 15, 2006)*
[19] Dongarra J. 2010 Performance of Various Computers Using Standard Linear Equations Software *University of Tennessee Computer Science Technical Report, CS-89-85, 2010*

MPI at Exascale

Rajeev Thakur,[1] Pavan Balaji,[1] Darius Buntinas,[1] David Goodell,[1] William Gropp,[2] Torsten Hoefler,[2] Sameer Kumar,[3] Ewing Lusk,[1] Jesper Larsson Traff[4]

[1]Argonne National Laboratory, Argonne, IL 60439, USA
[2]University of Illinois, Urbana, IL 61801, USA
[3]IBM Research, Yorktown Heights, NY 10598, USA
[4]Dept. of Scientific Computing, Univ. of Vienna, Austria

Abstract. With petascale systems already available, researchers are devoting their attention to the issues needed to reach the next major level in performance, namely, exascale. Explicit message passing using the Message Passing Interface (MPI) is the most commonly used model for programming petascale systems today. In this paper, we investigate what is needed to enable MPI to scale to exascale, both in the MPI specification and in MPI implementations, focusing on issues such as memory consumption and performance. We also present results of experiments related to MPI memory consumption at scale on the IBM Blue Gene/P at Argonne National Laboratory.

1. Introduction

We have already reached an era where the largest parallel machine in the world has close to 300,000 cores (the IBM Blue Gene/P at J¨ulich Supercomputing Center). Table 1 shows the top five largest parallel machines in terms of number of cores in the latest (June 2010) edition of the Top500 list [31]. These machines range in size from 150,000 to 300,000 cores. The most commonly used model for programming large-scale parallel systems today is MPI, and in fact, all the systems in Table 1 use MPI implementations that are derived from MPICH2 [23].

Although MPI runs successfully today on up to 300,000 cores, future extreme-scale systems are expected to comprise millions of cores. These systems may have several hundred thousand "nodes" (Figure 1), and each node itself may have large numbers (hundreds) of cores. The cores may comprise a mix of regular CPUs and accelerators such as GPUs, as illustrated in Figure 2. Many researchers and users wonder whether MPI will scale to systems of this size, or what is required to make scale MPI to this level. Furthermore, at exascale, MPI is likely to be used as part of a "hybrid" programming model

Table 1. Top five machines with the largest number of cores in the June 2010 Top500 list [31].

System	Location	Number of Cores
IBM BG/P	Jülich Supercomputing Center	294,912
Cray XT5	Oak Ridge National Laboratory	224,162
IBM BG/L	Lawrence Livermore National Laboratory	212,992
IBM BG/P	Argonne National Laboratory	163,840
IBM BG/P	Lawrence Livermore National Laboratory	147,456

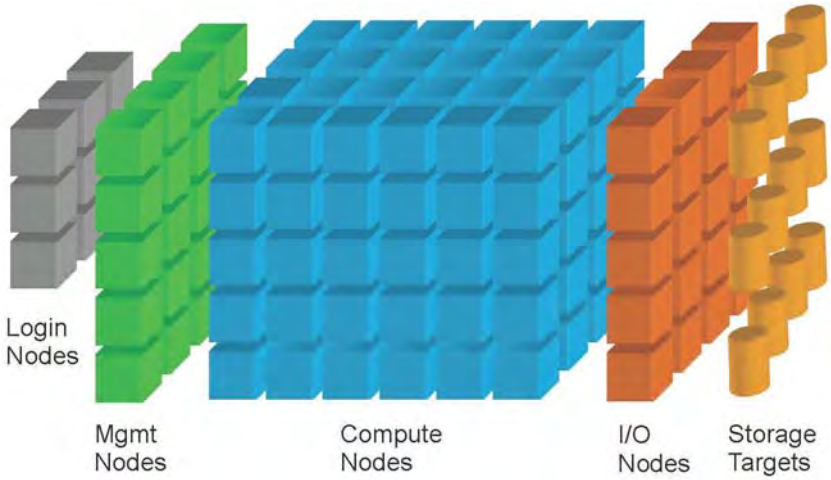


Figure 1. Future extreme-scale platforms with hundreds of thousands of nodes, each node with hundreds of cores.

(MPI+X), much more so than it is today. In such a model, MPI will be used to communicate between address spaces in conjunction with some other "shared memory" programming model (such as OpenMP, UPC, CUDA, or OpenCL) for programming within an address space. In other words, MPI and MPI implementations will need to support efficient hybrid programming. In this paper, we discuss the issues of scaling MPI to exascale, in terms of both what is needed in the MPI specification and what is needed from MPI implementations.

2. Scaling MPI to Exascale

Although the original designers of MPI were not thinking of exascale, MPI was always designed with scalability in mind. For example, a design goal was to enable implementations that maintain very little global state per process. Another design goal was to require very little memory management within MPI; all memory for communication can be in user space. MPI defines many operations as collective (called by a group of processes), which enables them to be implemented scalably and efficiently. Nonetheless, examination of the MPI standard reveals that some parts of the specification may need to be fixed for exascale. Section 3 describes these issues. Many of the issues are being addressed by the MPI Forum for MPI-3.

The main factors affecting MPI scalability are performance and memory consumption. A nonscalable MPI function is one whose time or memory consumption per process increase linearly (or worse) with the total number of processes. For example, if the memory consumption of `MPI_Comm_dup` increases linearly with the number of processes, it is not scalable. Similarly, if the time taken by `MPI_Comm_spawn` increases linearly or more with the number of processes being spawned, it indicates a nonscalable implementation of the function. Such examples need to be identified and fixed in the specification and in implementations. The goal should be to use constructs that require only constant space per process.

From an implementation perspective, the main requirement is that consumption of resources (memory, network connections, etc.) must not scale linearly with the number of processes. Since failures are expected to be common because of the large number of components, the implementation also needs to be resilient and tolerant to faults. Fault tolerance is needed from all levels of the stack—hardware, system software, and applications—and the MPI library must play its role. Since hybrid programming is likely to be common at exascale, the MPI implementation must also efficiently support concurrent communication from multiple threads of a process. In addition, the MPI

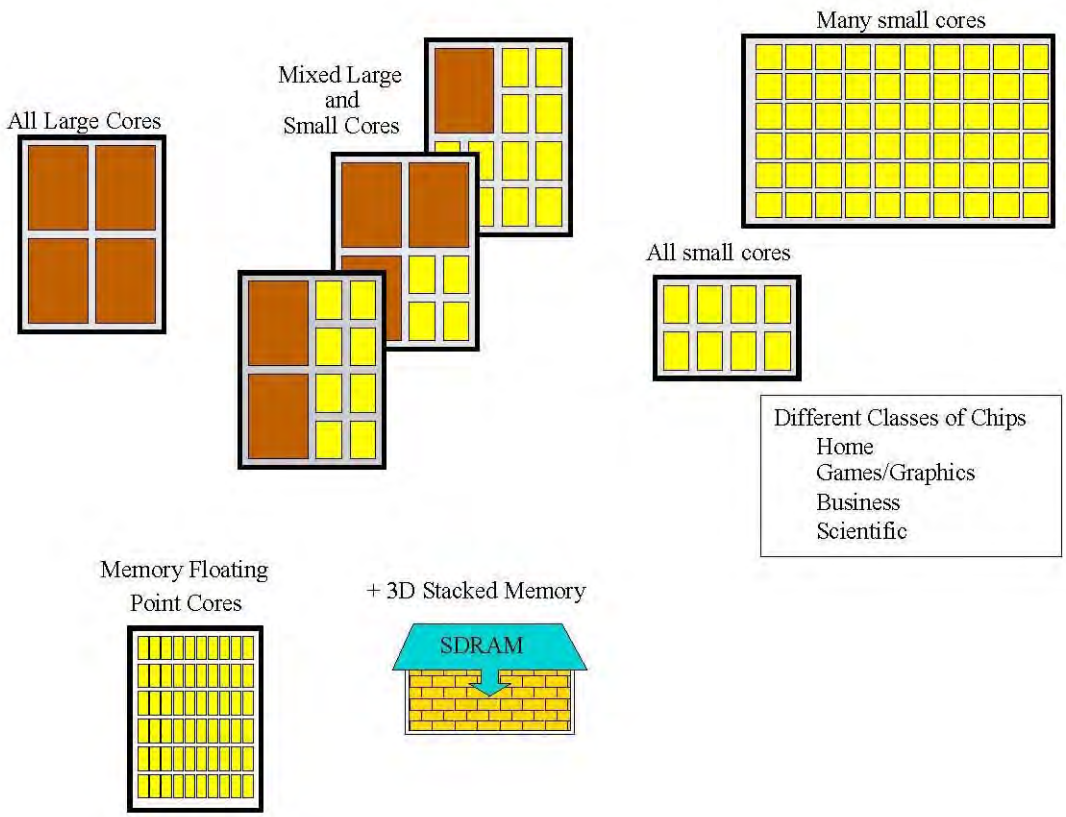


Figure 2. Many cores on a node.

implementation must provide excellent performance for the entire range of message sizes and for all MPI functions, not just simple latency and bandwidth benchmarks. In other words, there should be fewer performance surprises.

We discuss all these issues in further detail in the following sections.

3. Scalability Issues in the MPI Specification

Below we discuss aspects of the MPI specification that may have issues at extreme scale.

3.1. Sizes of Arguments to Some Functions

Some MPI functions take arguments that are arrays of size equal to the number of processes. An example is the irregular or "v" (vector) version of the collectives, such as `MPI_Gatherv` and `MPI_Scatterv`. These functions allow users to transfer different amounts of data among processes, and the amounts are specified by using an array of size equal to the number of processes. Using arrays that scale linearly with system size is nonscalable. An extreme example is the function `MPI_Alltoallw`, which takes six such arrays as arguments: counts, displacements, and datatypes for both send and receive buffers. On a million processes, each array will consume 4 MiB on each process (assuming 32-bit integers), requiring a total of 24 MiB just to call the function `MPI_Alltoallw` (i.e., just to pass the parameters to the function).

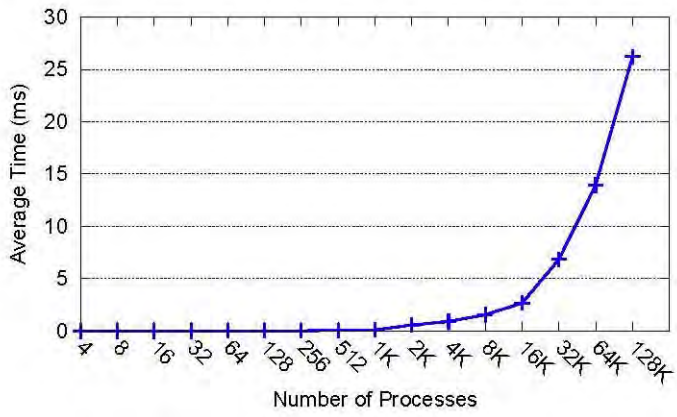


Figure 3. Time for doing a 0-byte alltoallv on IBM BG/P (no actual communication takes place).

Furthermore, an MPI implementation is forced to scan most of these arrays to determine what data needs to be communicated. On large numbers of processes, the time to read the entire array itself can be large, and it increases linearly with system size. This is particularly harmful in cases where a process needs to communicate with only a small number of other processes. For example, Figure 3 shows the time for a 0-byte `MPI_Alltoallv` on an IBM Blue Gene/P. No actual communication takes place since it is specified as 0 bytes. The time taken is just for each process to scan through the input array to determine that no communication is needed. As the number of processes increases, this time itself becomes significant. The MPI Forum is discussing ways to address this problem in MPI-3 [19], such as by defining sparse collective operations. A concrete proposal has been put forth in [13].

3.2. Graph Topology

MPI allows users to define virtual process topologies that express the communication pattern in the application. This feature in turn provides the implementation an opportunity to map the processes to the underlying system in a way that minimizes communication costs. Two types of virtual topologies are supported: a Cartesian topology and a general graph topology. The graph topology, which has existed since the first version of MPI, is one of the most nonscalable parts of the standard. It requires that each process specify the *entire* communication graph of *all* processes, not just its own communication information. The memory required for the purpose would clearly make it unusable on an exascale system. Other limitations of this interface are discussed in [32].

Thankfully, this problem has already been fixed in the latest version of the MPI Standard, MPI 2.2 [18]. Two new functions, `MPI_Dist_graph_create` and `MPI_Dist_graph_create_adjacent`, are defined that allow the user to specify the graph in a distributed, memory-efficient manner. However, the old interface is still available; and unless applications using the old interface make the effort to switch to the new interface, they will encounter scalability issues at large scale [12].

3.3. All-to-all Communication

MPI defines functions that allow users to perform all-to-all communication. All-to-all communication, however, is not a scalable communication pattern. Each process has a unique data item to send to every other process, which leads to limited opportunities for optimization compared with other collectives in MPI. This is not a problem with the MPI specification but is something that applications should be aware of and avoid as far as possible. Avoiding the use of all-to-all may require applications to use new algorithms.

3.4. Fault Tolerance

On exascale systems, the probability of failure or some other error in some part of the system is expected to be relatively high. As a result, greater resilience against failure is needed from all components of the software stack, from low-level system software to libraries and applications. The MPI specification already provides some support to enable users to write programs that are resilient to failure, given appropriate support from the implementation [11]. For example, when a process dies, instead of aborting the whole job, an implementation can return an error to any other process that tries to communicate with the failed process. The application then must decide what to do at that point.

However, more support from the MPI specification is needed for true fault tolerance. For example, the current set of error classes and codes needs to be extended to indicate process failure and other failure modes. Support is needed in areas such as detecting process failure, agreeing that a process has failed, rebuilding a communicator in the event of process failure or allowing it to continue to operate in a degraded state, and timing out for certain operations such as the MPI-2 dynamic process functions. A number of other researchers have studied the issue of fault tolerance in MPI in greater detail [4, 9, 14]. The MPI Forum is actively working on adding fault-tolerance capabilities to MPI-3 [20].

3.5. One-Sided Communication

Many applications have been shown to benefit from one-sided communication, where a process directly accesses the memory of another process instead of sending and receiving messages. For this reason, MPI-2 also defined an interface for one-sided communication that uses put, get, and accumulate calls and three different synchronization methods. This interface, however, has not been widely used for a number of reasons, the main being that its performance is often worse than regular point-to-point communication. The culprit is often the synchronization associated with one-sided communication. Another limitation is the lack of a convenient way to do atomic read-modify-write operations, which are useful in many parallel algorithms. Other issues with MPI one-sided communication are discussed in [3]. The MPI Forum is considering ways to fix these problems in MPI-3 [22].

3.6. Representation of Process Ranks

Another nonscalable aspect of MPI is the explicit use of lists of process ranks in some functions, such as the group creation routines `MPI_Group_incl` and `MPI_Group_excl`. While more concise representations of collections of processes are possible (for example, some group routines support ranges), the use of this sort of unstructured, nonscalable enumeration in some functions is problematic. Eliminating the explicit enumeration should be considered as an option for large scale.

4. MPI Implementation Scalability

MPI implementations must pay attention to two aspects as the number of processes is increased: memory consumption of any function and performance of *all* collective functions (not just the commonly optimized collective communication functions but also functions such as `MPI_Init` and `MPI_Comm_split`). We discuss some specific issues below.

4.1. Process Mappings

MPI communicators usually contain a mapping from MPI process ranks to processor ids. This mapping is often implemented as an array of size equal to the number of processes, which enables simple, constant-time lookup. Although convenient and fast, this solution is not scalable because it requires linear space per process per communicator and quadratic space over the system. To alleviate the problem, as a first step, communicators with the same process-to-processor mapping can share mappings. For example, if a communicator is duplicated with `MPI_Comm_dup`, the new communicator can share the mapping with the original communicator. (The MPICH2 implementation already does this.)

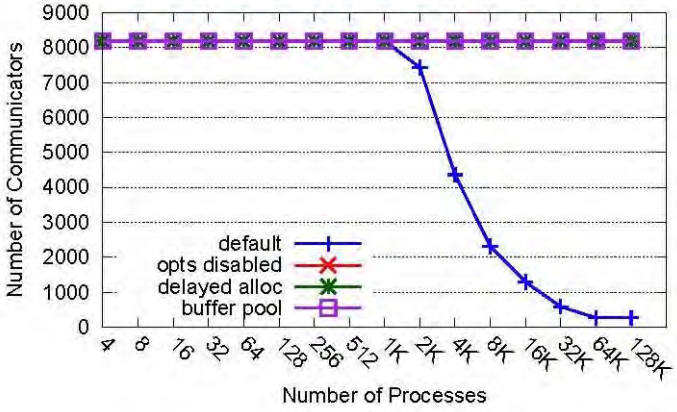


Figure 4. Maximum number of communicators that could originally be created with `MPI Comm dup` on IBM BG/P for varying numbers of processes.

In general, however, there is a need to explore more memory-efficient mappings, at least for common cases. Simple (and very restricted) solutions within the context of Open MPI were considered in [5]. A more general approach could be based on representations of mappings by simple linear functions, $ia + b \bmod p$. The identity mapping is often all that is needed for `MPI_COMM_WORLD`. Such linear representations, when possible, can be easily detected and cover many common cases, for example, subcommunicators that form consecutive segments from `MPI_COMM_WORLD`. A solution in this direction was explored in [34]. Other approaches, incorporated into the FG-MPI implementation, were described in [16]. However, this simple mapping covers only a small fraction of the $p!$ possible communicators, most of which cannot be represented by such simple means. For more general approaches to compact representations of mappings, see the citations in [34].

4.2. Creation of New Communicators

Creating duplicate communicators can consume a lot of memory at large scale if care is not taken. In fact, an application (Nek5000) running on the IBM Blue Gene/P at Argonne National Laboratory initially failed at only a few thousand processes because it ran out of memory after less than 60 calls to `MPI_Comm_dup`.

To study this issue, we ran a simple test that calls `MPI_Comm_dup` in a loop several times until it fails. We ran this test on the IBM BG/P and varied the number of processes. Figure 4 shows the results. Note that the maximum number of communicators supported by the implementation by default is 8,189 (independent of `MPI_Comm_dup`) because of a limit on the number of available context ids.

The number of new communicators that can be created drops sharply starting at about 2,048 processes. For 128K processes, the number drops to as low as 264. Although the MPI implementation on the BG/P does not duplicate the process-to-processor mapping in `MPI_Comm_dup`, it allocates some memory for optimizing collective communication. For example, it allocates memory to store "metadata" (such as counts and offsets) needed to optimize `MPI_Alltoall` and its variants. This memory usage is linear in `p`. Having such metadata per communicator is useful as it allows different threads to perform collective operations on different communicators in parallel. However, the per communicator memory usage increases with system size. Since the amount of memory per process is limited on the BG/P (512 MiB in virtual node mode), this optimization also limits the total number of communicators that can be created with `MPI_Comm_dup`.

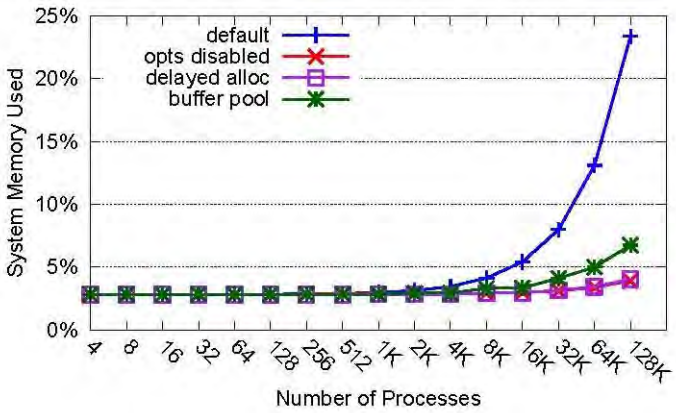


Figure 5. MPI memory usage on BG/P after 32 calls to `MPI Comm dup` of `MPI COMM WORLD` (in virtual node mode).

This scalability problem can be avoided in a number of ways. The simplest way is to use a BG/P environment variable to disable collective optimizations, which eliminates the extra memory allocation. However, it has the undesirable impact of decreasing the performance of all collectives. Another approach is to use an environment variable that delays the allocation of memory until the user actually calls `MPI_ Alltoall` on the communicator. This approach helps only those applications that do not perform `MPI_ Alltoall`.

A third approach, which we have implemented, is to use a buffer pool that is sized irrespective of the number of communicators created. Since the buffers exist solely to permit multiple threads to invoke `MPI_Alltoall` concurrently on different communicators, it is sufficient to have as many buffers as the maximum number of threads allowed per node, which on the BG/P is four. By using a fixed pool of buffers, the Nek5000 application scaled to the full system size without any problem.

Figure 5 shows the memory consumption in all these cases after 32 calls to `MPI_Comm_dup`. The fixed buffer pool enables all optimizations for all collectives and takes up only a small amount of memory.

4.3. Scalability of MPI Init

Since the performance of `MPI_Init` is not usually measured, implementations may neglect scalability issues in `MPI_Init`. On large numbers of processes, however, a nonscalable implementation of `MPI_Init` may result in `MPI_Init` itself taking several minutes. For example, on connection-oriented networks where a process needs to establish a connection with another process before communication, it is easiest for an MPI implementation to set up all-to-all connections in `MPI_Init` itself. This operation, however, involves $\Omega(p^2)$ amount of work and hence is nonscalable. A better approach is to establish no connections in `MPI_Init` and instead establish a connection when a process needs to communicate with another. This method does make the first communication more expensive, but only those connections that are really needed are set up. It also minimizes the number of connections, since applications written for scalability are not likely to have communication patterns where all processes directly communicate with all other processes.

Figure 6 shows the time taken by `MPI_Init` on a Linux cluster with TCP when all connections are set up eagerly in `MPI_Init` and when they are set up lazily. The eager method is clearly not scalable.

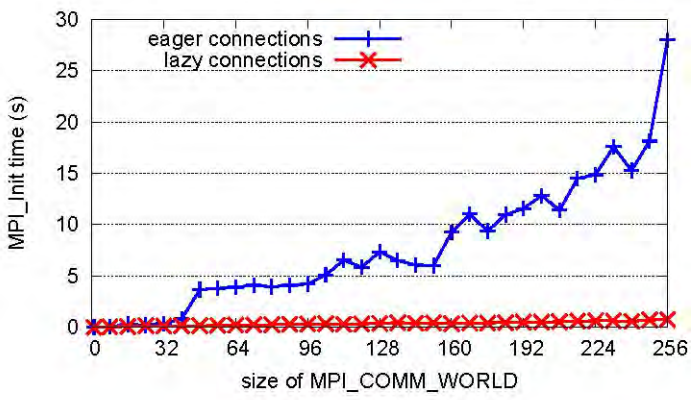


Figure 6. Time taken by MPI Init with eager versus lazy connections on an eight-core-per-node cluster using TCP.

4.4. Scalable Algorithms for Collective Communication

MPI implementations already use sophisticated algorithms for collective communication. Different algorithms are used depending on the message size: for short messages, they use an algorithm that minimizes latency; for long messages, they use algorithms that minimize bandwidth consumption. For example, broadcast is often implemented by using a binomial tree for short messages and a scatter-allgather for long messages [6, 30]. The scatter-allgather may become inefficient at large scale because the scatter may result in blocks that are too small. For example, for a 1 MiB broadcast on one million processes, the scatter phase will result in blocks of size 1 byte. Such problems can be countered by using hybrid algorithms that first do a logarithmic broadcast to a subset of nodes and then a scatter/allgather on many subsets at the same time [33]. Algorithms with similar properties for reduction operations are given in [26].

Global collective acceleration supported by many networks such as Quadrics, InfiniBand, and Blue Gene may be another solution for collectives on `MPI_COMM_WORLD`. On the Blue Gene/P, for example, the `MPI_Broadcast`, `MPI_Reduce`, `MPI_Allreduce`, `MPI_Scatter`, `MPI_Scatterv`, and `MPI_ Allgather` collectives take advantage of the combine and broadcast features of the tree network [2].

4.5. Enabling Hybrid Programming

Exascale machines are expected to have many more cores per node than today, but the memory per core is likely to remain the same. Applications, however, want to access more and more memory from each process. A solution to this problem is to use a combination of shared-memory and message-passing programming models (MPI+X): MPI for moving data between address spaces and some shared-memory model (X) for accessing data within an address space that spans multiple cores or even multiple nodes. Options for X include the following.

- OpenMP. This option has been well studied [15, 17, 25, 27]. One limitation is that the shared memory is usually restricted to the address space of a single physical node.
- PGAS languages such as UPC [8] or CoArray Fortran [24]. This option is not as well understood, although there has been some recent preliminary work with UPC [7]. The advantage of this approach is that the shared address space can span the memories of multiple nodes.
- CUDA/OpenCL. This option is needed for GPU-accelerated systems.

What makes hybrid programming possible is the carefully defined thread-safety semantics of MPI [10]. This specification does not require a particular threads implementation or library, such as Pthreads or OpenMP. Rather, it defines a mechanism by which the user can request the desired level of

thread safety and the implementation can indicate the level of thread safety that it provides. This mechanism allows the implementation to avoid incurring the cost of providing a higher level of thread safety than the user needs.

The MPI Forum is exploring further enhancements to MPI to support efficient hybrid programming [21]. An interesting proposal being discussed is to extend MPI to support multiple communication endpoints per process [28]. The basic idea is as follows. In MPI today, each process has one communication endpoint (rank in `MPI_COMM_WORLD`). Multiple threads communicate through that one endpoint, requiring the implementation to do use locks, which turn out to be expensive [29]. This proposal allows a process to define multiple endpoints. Threads within a process attach to different endpoints and communicate through those endpoints as if they are separate ranks. The MPI implementation can avoid using locks if each thread communicates on a separate endpoint.

4.6. Fewer Performance Surprises

Sometimes we hear the following from users:

- "I replaced `MPI_Allreduce` by `MPI_Reduce + MPI_Bcast` and got better results."
- "I replaced `MPI_Send(n)` by `MPI_Send(n/k) + MPI_Send(n/k) + ... + MPI_Send(n/k)` and got better results."
- "I replaced `MPI_Bcast(n)` by my own algorithm for broadcast using `MPI_Send(n)` and `MPI_ Recv(n)` and got better results."

None of these situations should happen. If there is an obvious way intended by the MPI standard of improving communication time, a sound MPI implementation should do so, and not the user! We refer to these as performance surprises.

Although MPI is portable, there is considerable performance variability among MPI implementations— lots of performance surprises. In [35], we defined a set of common-sense, self-consistent performance guidelines for MPI implementations, for instance, the following:

- Subdividing messages into multiple messages should not reduce the communication time. For example, an `MPI_Send` of 1500 bytes should not be slower than two calls to `MPI_Send` for 750 bytes each.
- Replacing an MPI function with a similar function that provides additional semantic guarantees should not reduce the communication time. For example, `MPI_Send` should not be slower than `MPI_ Ssend`.
- Replacing a specific MPI operation by a more general operation by which the same functionality can be expressed should not reduce communication time. For example, `MPI_Scatter` should not be slower than `MPI_Bcast` (Figure 7).

Nonetheless, we found instances where such simple requirements are violated. For example, on the IBM BG/P, scatter is about four times slower than broadcast, as Figure 8 shows. The reason is because broadcast is implemented by using the hardware support provided by the interconnection network, whereas scatter is implemented in software. However, it should be possible to implement scatter by doing a broadcast and discarding the unnecessary data and achieve four times better performance. It would, of course, need extra memory allocation in the implementation.

Figure 7. Scatter should be faster, or at least no slower, than a broadcast because it does less work.

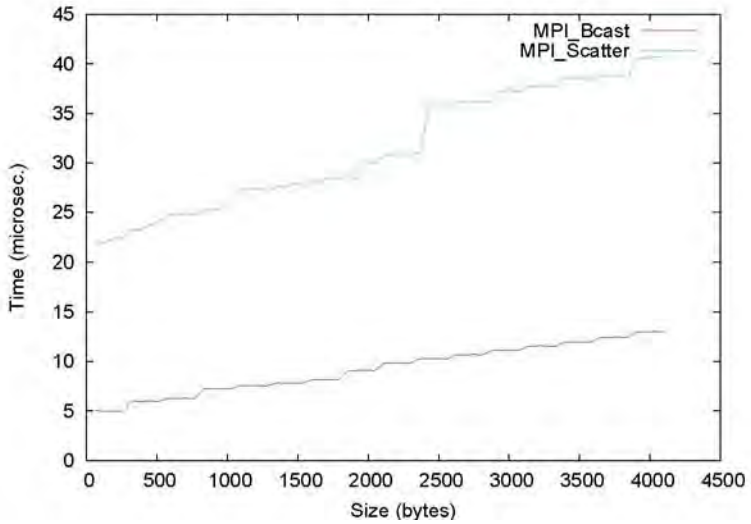


Figure 8. Performance of MPI Scatter versus MPI Bcast on IBM BG/P.

A similar performance surprise is encountered in most MPI implementations because of the switch from eager to rendezvous protocol at a particular message size. Short messages are sent eagerly to the destination assuming that there is enough memory available to store them. At some message size, the implementation switches to a rendezvous protocol, where it waits for an acknowledgment from the destination that the matching receive has been posted and hence memory is available to store in the incoming message. This often leads to a performance graph similar to the one in Figure 9, where there is a sharp jump in communication time when the rendezvous threshold is crossed. In this example, a user could get better performance for a 1500-byte message by sending it as two 750-byte messages instead.

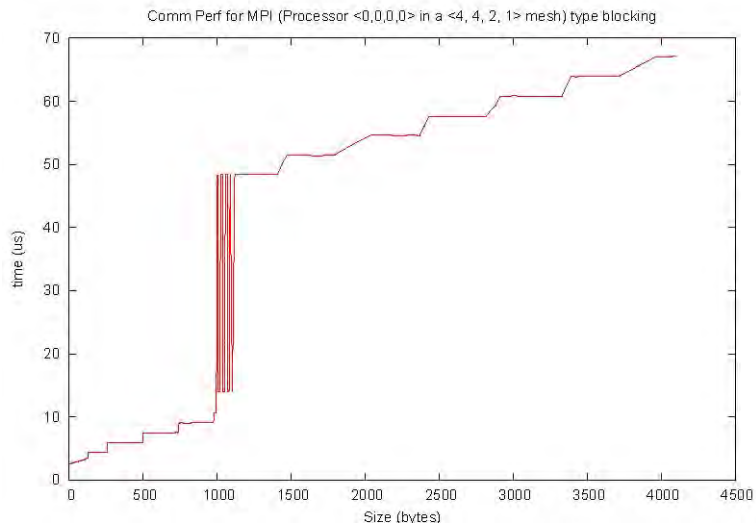


Figure 9. Latency on IBM BG/L. The large jump at 1024 bytes is due to change of protocol.

If care is not taken, such performance surprises will be even more common at exascale because of the large numbers of processes and unexpected interactions among communication patterns at large scale. Tools need to be written to check for these requirements, which would help implementers identify and fix problems.

5. Conclusions

Exascale systems are expected to be available in less than a decade. Although MPI runs successfully on today's petascale systems, for it to run efficiently on exascale systems with millions of cores, some issues need to be fixed both in the MPI specification and in MPI implementations. These issues are related primarily to memory consumption, performance, and fault tolerance. At small scale, memory consumption is often overlooked, but it becomes critical at large scale particularly because the amount of memory available per core is not expected to increase. The MPI Forum, which is currently meeting regularly to define the next version of MPI (MPI-3), is addressing several of the issues that must be fixed in the MPI specification. Correspondingly, MPI implementations are also addressing the issues that they must fix in order to scale to exascale. As a result of these efforts, we believe MPI will run successfully on exascale systems when they are available. Further details on issues related to scaling MPI to millions of cores can be found in [1].

Acknowledgments

We thank the members of the MPI Forum who participated in helpful discussions of the presented topics. We also thank the anonymous reviewers for comments that improved the manuscript. This work was supported in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DEAC02-06CH11357 and award DE-FG02-08ER25835, and in part by the National Science Foundation award 0837719.

References

[1] Pavan Balaji, Darius Buntinas, David Goodell, William Gropp, Torsten Hoefler, Sameer Kumar, Ewing Lusk, Rajeev Thakur, and Jesper Larsson Traff. MPI on millions of cores. *Parallel Processing Letters*, 2011. To appear.

[2] Pavan Balaji, Anthony Chan, Rajeev Thakur, William Gropp, and Ewing Lusk. Toward message passing for a million processes: Characterizing MPI on a massive scale Blue Gene/P. *Computer Science – Research and Development*, 24(1–2):11–19, 2009.

[3] Dan Bonachea and Jason Duell. Problems with using MPI 1.1 and 2.0 as compilation targets for parallel language implementations. In *2nd Workshop on Hardware/Software Support for High Performance Sci. and Eng. Computing*, 2003.

[4] George Bosilca, Aurelien Bouteiller, Franck Cappello, Samir Djilali, Gilles Fedak, Cedile Germain, Thomas Herault, Pierre Lemarinier, Oleg Lodygensky, Frederic Magniette, Vencent Neri, and Anton Selikhov. MPICH-V: Toward a scalable fault tolerant MPI for volatile nodes. In *Proc. of SC 2002*. IEEE, 2002.

[5] Mohamad Chaarawi and Edgar Gabriel. Evaluating sparse data storage techniques for MPI groups and communicators. In *Computational Science. 8th International Conference (ICCS)*, volume 5101 of *Lecture Notes in Computer Science*, pages 297–306. Springer-Verlag, 2008.

[6] Ernie Chan, Marcel Heimlich, Avi Purkayastha, and Robert A. van de Geijn. Collective communication: Theory, practice, and experience. *Concurrency and Computation: Practice and Experience*, 19(13):1749–1783, 2007.

[7] James Dinan, Pavan Balaji, Ewing Lusk, P. Sadayappan, and Rajeev Thakur. Hybrid parallel programming with MPI and Unified Parallel C. In *Proceedings of the ACM International Conference on Computing Frontiers*, May 2010.

[8] Tarek El-Ghazawi, William Carlson, Thomas Sterling, and Katherine Yelick. *UPC: Distributed Shared-Memory Programming*. Wiley-Interscience, May 2005.

[9] Graham E. Fagg and Jack J. Dongarra. FT-MPI: Fault tolerant MPI, supporting dynamic applications in a dynamic world. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface. 7th European PVM/MPI Users' Group Meeting*, volume 1908 of *Lecture Notes in Computer Science*, pages 346–353. Springer-Verlag, 2000.

[10] William Gropp and Rajeev Thakur. Thread safety in an MPI implementation: Requirements and analysis. *Parallel Computing*, 33(9):595–604, 2007.

[11] William D. Gropp and Ewing Lusk. Fault tolerance in MPI programs. *Int'l Journal of High Performance Computer Applications*, 18(3):363–372, 2004.

[12] Torsten Hoefler, Rolf Rabenseifner, Hubert Ritzdorf, Bronis R. de Supinski, Rajeev Thakur, and Jesper Larsson Tr¨aff. The scalable process topology interface of MPI 2.2. *Concurrency and Computation: Practice and Experience*, 2010. To appear.

[13] Torsten Hoefler and Jesper Larsson Tr¨aff. Sparse collective operations for MPI. In *Proc. of 14th Int'l Workshop on High-level Parallel Programming Models and Supportive Environments at IPDPS*, 2009.

[14] Hideyuki Jitsumoto, Toshio Endo, and Satoshi Matsuoka. ABARIS: An adaptable fault detection/recovery component framework for MPIs. In *Proc. of 12th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems (DPDNS '07) in conjunction with IPDPS 2007*, 2007.

[15] Gabriele Jost, Haoquiang Jin, Dieter an Mey, and Ferhat F. Hatay. Comparing the OpenMP, MPI and hybrid programming paradigms on an SMP cluster. In *Proceedings of EWOMP'03*, 2003. http: //www.nas.nasa.gov/News/Techreports/2003/PDF/nas-030019.pdf.

[16] Humaira Kamal, Seyed M. Mirtaheri, and Alan Wagner. Scalability of communicators and groups in MPI. In *ACM International Symposium on High Performance Distributed Computing (HPDC)*, 2010.

[17] Ewing Lusk and Anthony Chan. Early experiments with the OpenMP/MPI hybrid programming model. In Rudolf Eigenmann and Bronis R. de Supinski, editors, *OpenMP in a New Era of Parallelism*, volume 5004 of *Lecture Notes in Computer Science*, pages 36–47. Springer, 2008. IWOMP, 2008.

[18] MPI Forum. MPI: A Message-Passing Interface Standard. Version 2.2, September 4th 2009. available at: http://www.mpi-forum.org (July 2010).

[19] MPI Forum Collectives Working Group (Lead: Torsten Hoefler). http://meetings.mpi-forum.org/mpi3.0 collectives.php (July 2010).
[20] MPI Forum Fault Tolerance Working Group (Lead: Rich Graham). http://meetings.mpi-forum.org/mpi3.0 ft.php (July 2010).
[21] MPI Forum Hybrid Programming Working Group (Lead: Pavan Balaji). http://meetings.mpi-forum.org/mpi3.0 hybrid.php (July 2010).
[22] MPI Forum RMA Working Group (Leads: William Gropp and Rajeev Thakur). http://meetings.mpi-forum.org/mpi3.0 rma.php (July 2010).
[23] Mpich2. http://www.mcs.anl.gov/mpi/mpich2.
[24] R. W. Numrich and J. K. Reid. Co-Array Fortran for parallel programming. *ACM Fortran Forum*, 17(2):1–31, August 1998.
[25] Rolf Rabenseifner, Georg Hager, and Gabriele Jost. Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes. In *Proc. of 17th Euromicro Int'l Conference on Parallel, Distributed, and Network-Based Processing (PDP 2009)*, pages 427–236, 2009.
[26] Rolf Rabenseifner and Jesper Larsson Tr¨aff. More efficient reduction algorithms for message-passing parallel systems. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface. 11th European PVM/MPI Users' Group Meeting*, volume 3241 of *Lecture Notes in Computer Science*, pages 36–46. Springer-Verlag, 2004.
[27] Ashay Rane and Dan Stanzione. Experiences in tuning performance of hybrid MPI/OpenMP applications on quad-core systems. In *Proc. of 10th LCI Int'l Conference on High-Performance Clustered Computing*, 2009.
[28] Marc Snir. MPI-3 hybrid programming proposal, version 7. http://meetings.mpi-forum.org/mpi3.0_hybrid.php.
[29] Rajeev Thakur and William Gropp. Test suite for evaluating performance of multithreaded MPI communication. *Parallel Computing*, 35(12):608–617, December 2009.
[30] Rajeev Thakur, Rolf Rabenseifner, and William Gropp. Optimization of collective communication operations in MPICH. *Int'l Journal of High-Performance Computing Applications*, 19(1):49–66, 2005.
[31] Top500 list. http://www.top500.org/lists/2010/06, June 2010.
[32] Jesper Larsson Tr¨aff. SMP-aware message passing programming. In *Proc. of 8th Int'l Workshop on High-level Parallel Programming Models and Supportive Environments at IPDPS*, pages 56–65, 2003.
[33] Jesper Larsson Tr¨aff. A simple work-optimal broadcast algorithm for message-passing parallel systems. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface. 11th European PVM/MPI Users' Group Meeting*, volume 3241 of *Lecture Notes in Computer Science*, pages 173–180. Springer-Verlag, 2004.
[34] Jesper Larsson Tr¨aff. Compact and efficient implementation of the MPI group operations. In *Recent Advances in Message Passing Interface. 17th European MPI Users' Group Meeting*, volume 6305 of *Lecture Notes in Computer Science*, pages 170–178. Springer-Verlag, 2010.
[35] Jesper Larsson Tr¨aff, William D. Gropp, and Rajeev Thakur. Self-consistent MPI performance guidelines. *IEEE Transactions on Parallel and Distributed Systems*, 21(5):698–709, 2010.

Coupled whole device simulations of plasma transport in tokamaks with the FACETS code

A H Hakim,[1] J R Cary,[1] J Candy,[2] J Cobb,[3] R H Cohen,[4] T Epperly,[4] D J Estep,[5] S Krasheninnikov,[6] A D Malony,[7] D C McCune,[8] L McInnes,[9] A Pankin,[10] S Balay,[9] J A Carlsson,[1] M R Fahey,[3] R J Groebner,[2] S E Kruger,[1] M Miah,[1] A Pletzer,[1] S Shasharina,[1] S Vadlamani,[1] D Wade-Stein,[1] T D Rognlien,[4] A Morris,[7] S Shende,[7] G W Hammett,[8] K Indireshkumar,[7] A Yu Pigarov,[6] H Zhang[9]

[1]Tech-X Corporation, 5621 Arapahoe Avenue, Suite A, Boulder, CO 80303
[2]General Atomics
[3]Oak Ridge National Laboratory
[4]Lawrence Livermore National Laboratory
[5]Colorado State University
[6]University of California at San Diego
[7]ParaTools, Inc.
[8]Princeton Plasma Physics Laboratory
[9]Argonne National Laboratory
[10]Lehigh University

E-mail: ammar@txcorp.com

Abstract. The FACETS project aims to provide computational tools for whole device simulation of tokamak transport for use in fusion applications. The framework provides flexibility by allowing users to choose the best model for a given physics target. Our goals are to develop accurate transport solvers using neoclassical and turbulent fluxes with varying degree of fidelity and computational complexity, including embedded gyrokinetic models. Accurate sources using both ICRH wave absorption and neutral beam injection, using parallel source components, are included. Modeling of the plasma edge using a fluid based component, UEDGE, is performed and coupled to the core solver. The core region is simulated using a newly developed parallel, nested iteration based nonlinear solver while the UEDGE uses nonlinear solves from the PETSc/SNES solver package. As a first application we present coupled core–edge simulations of pedestal buildup in the DIIID tokamak.

1. Introduction

Fusion promises to be an efficient source of clean energy. The world-wide fusion community is embarking on an ambitious next step: the ITER project (http://www.iter.org/). ITER is a joint international research and development project that aims to demonstrate the scientific and technical feasibility of fusion power. With over 16 billion Euros investment planned in ITER from the EU, the US, Russia, Japan, China, and India, ITER is an important next step for the worldwide fusion program, and an important part of the U.S. Department of Energy mission. For this reason, ITER was listed as

the highest priority in the twenty-year outlook of DOE facilities [1]. The success of ITER requires that the facility is optimally used, which not only means achieving the best operating conditions, but also achieving a thorough understanding of the experimental results.

To obtain maximal returns on this investment it is critical to have computational tools to help optimize the design and provide theoretical insights into experimental results. Several such computational efforts are currently underway, some devoted to studying detail physical processes in the plasma and others devoted to performing whole device simulations using coupled reduced physics models. The FACETS (Framework Application for Core-Edge Transport Simulations) project [2] has the goal of providing whole-tokamak modeling through coupling separate components for each of the core region, edge region, and wall, with realistic sources. This is a complex problem, as each component is parallel in its own right, and each can be made parallel in a distinct manner.

Given the large disparity in space and time scales in a tokamak, progress can be achieved only by separating physics into different parts, such that for each part, valid, reduced approximations exist. For example, in the core of the plasma, the rapid transport along field lines assures that the plasma parameters such as density and temperature are, over long time scales, constant on toroidally nested flux surfaces. This reduces the transport equation to one dimension for evolution on the discharge time scale. However, the cross-field transport coefficient are difficult to determine accurately and sophisticated models, with varying predictive capability, have been developed. The current state-of-art in core-transport modelling uses gyrokinetic turbulence calculation to account for the dominant anomalous transport processes. Although this "embedded" gyrokinetic approach is computationally expensive it is feasible with current generation of Leadership Class Facilities (LCFs). In the plasma edge, though, simulations must be global and two-dimensional. The separation of scales in the edge is not as clean as in the core region and for this reason global edge turbulence models need to be developed to achieve true predictive capability. The development of such edge turbulence models, even using fluid approximations, is still an active research area.

The complexity of whole-device simulations of tokamaks translates to requiring a software component approach, as FACETS is taking. FACETS is bringing together successively more accurate and, hence, computationally demanding components to model the complete plasma device. It is being developed to run on LCFs to be able to use the most computationally demanding components, while at the same time it is usable on laptops for less demanding models. FACETS is constructing a C++ framework for incorporating the best software packages and physics components. A description of the FACETS framework is given in [3] in which the component approach and description of key concepts in the framework is given.

In this paper we present results from coupled core-edge simulations of pedestal buildup in the DIII-D tokamak for a selected shot. The rest of the paper is organized as follows. We first describe the core and edge equations used in the analysis and also describe the coupling scheme to advance the coupled system self-consistently in time. Then we describe the shot (DIII-D shot 118897) and initial conditions and edge interpretive analysis used in the calculations. Results are then presented and discussed. We then make some concluding remarks and indicate further work on modelling the core-edge physics more accurately.

2. Core and Edge Models and Coupling Scheme

High-performance tokamak plasma discharges (so-called "H-mode" plasmas) are characterized by steep gradients of temperature and density near the plasma separatrix where open field lines separate from closed field lines. This region is referred to as the "H-mode pedestal." Experiments and core transport simulations indicate that core profiles are relatively insensitive to the overall values of density and temperature, but that the shape is set by stiff transport due to kinetic turbulence. Thus, the overall fusion gain depends sensitively on the parameters at the top of the pedestal temperature and density [4]. The pedestal height serves as a boundary condition to the core plasma region. The physics determining the formation of the pedestal (the so-called L-H transition) is not fully understood, and is the subject of active investigations [5]. As a first test of the FACETS code we have undertaken the

simulation of the the pedestal buildup in a particular shot of the DIII-D tokamak. However, due to poor understanding of the edge physics and lack of availability of predictive models for the edge transport we have used an interpretive analysis to determine the edge cross-field transport coefficients. These are held constant throughout the simulation. Although not fully self-consistent, this serves to test the infrastructure and also gives insight into the time-scales over which edge transport coefficients evolve.

Well inside the separatrix the plasma is hot which leads to large disparities in the transport parallel and perpendicular to the field lines. This in turn implies that, on transport time-scales, the plasma equilibriates in the poloidal direction and the physics can now be described as a set of one dimensional equations for the perpendicular transport of density and energy

$$\frac{\partial n}{\partial t} + \frac{1}{V'}\frac{\partial}{\partial \rho}(V' F_n^\rho) = S_n \tag{1}$$

$$\frac{3}{2}\frac{\partial}{\partial t}(nT_s) + \frac{1}{V'}\frac{\partial}{\partial \rho}(V' F_s^\rho) = S_s \tag{2}$$

where $n(\rho, t)$ is the plasma number density, $T_s(\rho, t)$ are electron ($s = e$) and ion ($s = i$) temperatures, F_n^ρ is the contravariant particle flux, F_s^ρ are the contravariant thermal fluxes and S_n (ρ, t) and S_s (ρ, t) are the particle and thermal sources. The independent variable is the dimensionless normalized toroidal flux ρ and the radial geometry is encapsulated in $V' \equiv dV/d\rho$, where $V(\rho)$ is the volume enclosed inside the flux surface ρ.

The fluxes are computed using a combination of anomalous and neoclassical terms $F_{GLF} + F_{CH} + F_{ETG}$, where F_{GLF} is the anomalous flux computed using the GLF23 [6] model, F_{CH} is the neoclassical flux computed using the Chang-Hinton [7] model and F_{ETG} is anomalous electron transport due to Horton-ETG model [8]. The system of core transport equations Eq. (1) and Eq. (2) are highly non-linear and stiff due to the strong dependence of the fluxes on the gradients of the density and temperatures. A nested-iterations based implicit solver is implemented in FACETS to evolve this set of equations [9].

As we approach the separatrix the plasma cools and the field lines are no longer nested. This leads to two-dimensional effects and the plasma transport both along and perpendicular to the field lines now needs to be evolved. FACETS incorporates the fluid edge code UEDGE [10,11] to solve the edge plasma equations. UEDGE includes full single-or double-null X-point geometry with a simulation domain including the region spanning well inside the separatrix and extending to the outer wall and the divertor plate region. A reduced set of the Braginskii transport equations is solved. Usually the cross diffusion terms are turned off and the Alfven waves are suppressed by not evolving the induction equation for the magnetic field. Additional assumptions are used to express the equations in a form more amenable to disabling various physics terms to allow faster solution times, or to allow better physics understanding. The specific form of the equations are found in Ref. [11].

The coupling between the core and the edge regions is performed using an explicit scheme. There are several challenges in setting up an self-consistent problem across the one-dimensional predictive core region with the two-dimensional interpretive edge region. First, the initial conditions need be to C^1 continuous across the core-edge (CE) interface. Second, the transport coefficients at the CE interface need to match. This is particularly difficult as completely different models are used to compute the fluxes in the core and edge solvers. An ad-hoc scheme is presently implemented in FACETS to enforce this. The continuity in the initial conditions is ensure by using the same set of experimental data to initialize both the core and the edge. However, this does not guarantee the continuity of slopes at the CE interface as two components use different grids and discretization schemes.

The coupled simulation is run as follows. The core and edge components are run concurrently for a specified time-step dt. Typically, both component need to sub-cycle to evolve their solutions by this time-step. The particle and energy fluxes are passed from the core to the edge, i.e., $F^\rho V' / A = F^\rho / <|$

∇p | > is sent to the edge, where A is the area of the flux surface at the CE interface and the angular brackets indicate flux surface averaging. The flux-surface averaged temperature and number density are passed from the edge to the core. These steps are repeated and the simulation evolved in time.

The explicit coupling scheme described above is stable as long as sufficiently small time-steps are taken. Numerical experiments show that a time-step on the order of several 100 μs can be taken without the coupled solution going unstable. A fully implicit coupling scheme, although not used in this paper, is also implemented in FACETS and presently being tested for use in core-edge simulations.

3. Problem Setup and Results

Coupled core-edge simulations using the methods described in the previous section are performed to study the pedestal buildup in the DIII-D tokamak. For this we have chosen to simulate the time slice from 1555 to 1590 ms of DIII-D shot 118897. From the experimental wave forms it is seen that the neutral beams are turned on around 1490 ms and the plasma density increases from around 2.8×10^{19} m^3 to 3.9×10^{19} m^3.The beam power in the chosen interval averages to around 4.5 MW and eventually drops down to 2.5 MW around 1600 ms. The discharge is ELM free till around 2355 ms.

Beam sources are held constant during the simulation and are taken from an interpretive ONETWO simulation. See Figure 1 for beam source heating profiles for ions and electrons. The simulation is initialized using experimental data averaged around 1555 ms into the discharge. See Figure 2 for initial density and temperature profiles. The core component is run to $\rho = 0.85$ and the edge component from there to the wall. Note that this puts the pedestal inside the edge component. The reason for doing this is twofold: (a) the core transport models do not work very well for $\rho > 0.85$ and, (b) we want to ensure that the edge is sufficiently inside the separatrix for the solutions to be relaxed along the poloidal direction.

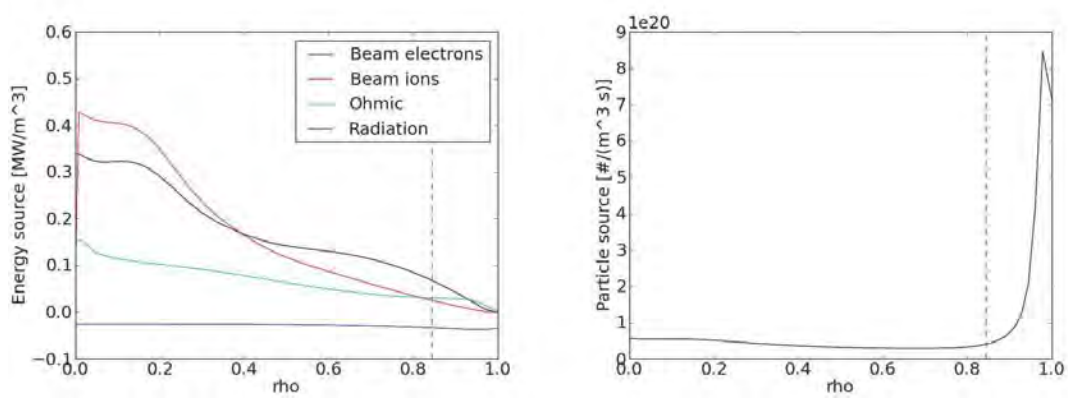


Figure 1. Power sources for electrons and ions (left) and plasma particle source (right). Total beam power for ions is 1.78 MW, for electrons 2.21 MW and Ohmic heating is 0.93 MW and radiation loss is 0.40 MW. Net particle injection rate is 2.12×10^{21} s^{-1}. Out of these, the fraction of particles into the core is 4.99×10^{20} s^{-1}. Rest (1.617×10^{21} s^{-1}) should be accounted for in the edge.

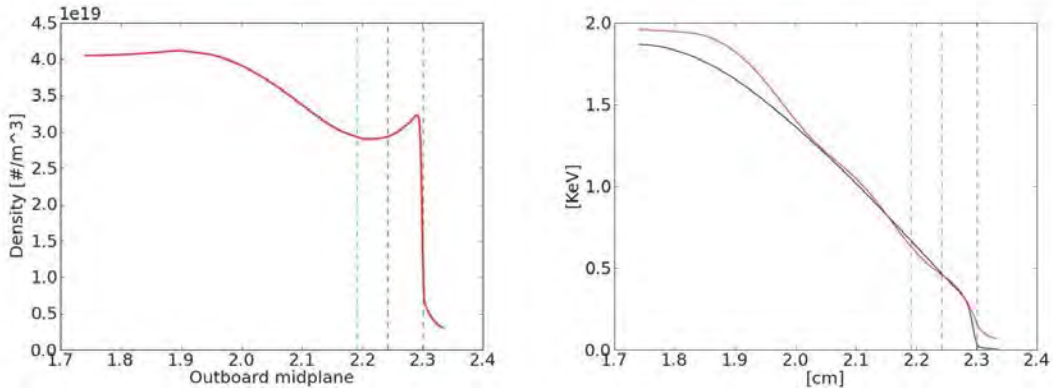


Figure 2. Initial density (left) and temperature (right) profiles for electrons (black) and ions (red) for shot 118897 at 1555 ms. The blue dashed lines indicate the core-edge interface, the green dashed line the separatrix and the cyan line the start of transition from predictive to interpretive fluxes.

An initial set of calculations using experimental profiles and sources is performed to determine the cross-field diffusivities for UEDGE. See Figure 3. The diffusivites drop down significantly in the pedestal to create a transport barrier for the pedestal formation. As mentioned before this procedure is not self-consistent as the core is predictive, but lacking predictive edge models this method must be adopted. A fully interpretive edge model should predict a similar transport barrier. Once these diffusivities were determined they were held constant through the 35 ms simulation time.

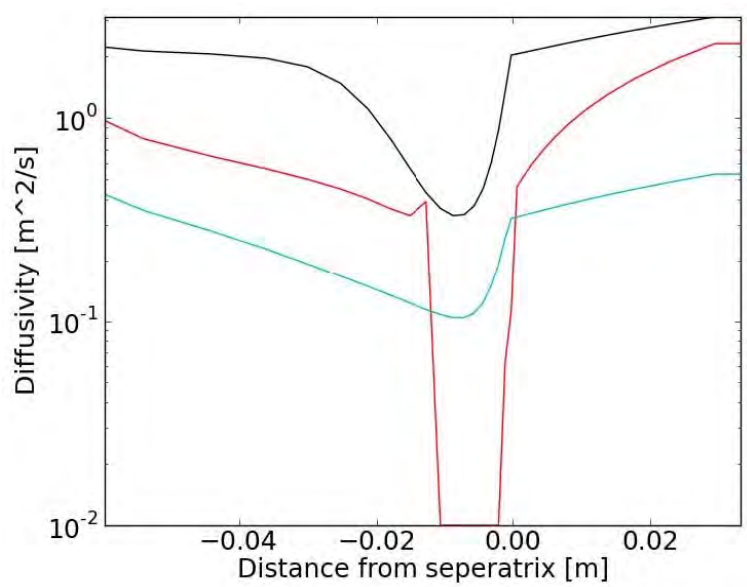


Figure 3. UEDGE diffusivities from interpretive calculations. Red line D, black line χe, blue line χi.

The core and edge components were evolved using the explicit coupling scheme described above. A time-step of 200 μs was used to couple the components, although the components internally were allowed to choose their own time-steps. Plasma density, electron and ion energy equations were evolved. See Figure 4 for comparison between simulation results and experimental profiles averaged around 1590 ms into the discharge. Our initial results show that the electron temperature is predicted better that the ion temperature. As seen in the figure the ion pedestal temperature rises beyond that

seen in the experiment, indicating a greater flow of ion thermal energy into the edge. This leads to a corresponding loss in ion energy near the magnetic axis. This is behaviour is being investigated and it is believed that a proper accounting for the $\omega_{|E\times B|}$ shear profile will lead to a damping to the ion thermal flux near the CE interface and improve the ion temperature predictions. It is also seen that the density buildup in the edge is under-predicted. The reason for this could be that the neutral particle source fueling the edge is not sufficient to supply the net line-integrated density in the tokamak. An additional gas-puff source would supply this particle source and this is being investigated at present.

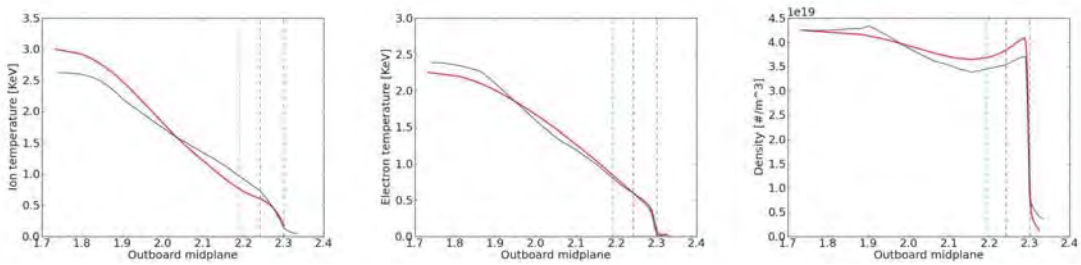


Figure 4. Final experimental (red) and simulation results (black) for ion (left) and electron (middle) temperatures at 35 ms. Final experimental density profile (red) and simulation results (black) are shown in the right panel. The ion temperature is over-predicted at the pedestal while the plasma density is under-predicted. Tuning the gas puff to supply an additional particle fueling source could lead to a higher density buildup in the edge.

4. Conclusions

We have presented coupled core-edge simulations of pedestal buildup in a selected shot of the DIII-D tokamak. Our initial simulations show that FACETS can evolve the plasma density in addition to the electron and ion energy equations self-consistently using coupled core-edge simulations. Our initial calculations have shown a sensitivity of the pedestal buildup on the core-edge power balance and also to the presence of a gas-puff as a fueling source in the edge. These calculations show the ability of the framework to perform concurrent coupled simulations with multiple components and serve as a test for the explicit coupling algorithms. We expect improvements in the predicted ion temperature and plasma density with a better accounting of the shear flow profiles and a proper tuning of the gas-puff. We believe that our agreement with experiments is reasonable given the uncertainties in the core density model (GLF23) and the lack of a predictive edge model.

Our current work in FACETS is to increase the physics fidelity of the simulations. We have incorporated the NUBEAM neutral beams package that allows the self-consistent computation of the beam deposition into the core. However, before NUBEAM can be used on a production level it will have to be extended to deposit power into the edge. We have incorporated the TGLF[12] anomalous transport model to better predict the core transport coefficients. We have implemented a Newton-method based implicit coupling algorithm that we expect will allow us to take 10 ms time-steps, thus increasing the discharge time that can be simulated. We have completed the implementation of a sophisticated wall model that will be incorporated into the simulations. FACETS is now being used to simulate other DIII-D discharges to better quantify the core-edge physics.

References

[1] U. D. of Energy, “Facilities for the future of science: A twenty-year outlook.” http://www.er.doe.gov/about/ Future/Facilities\for\the\Future\of\Science.htm, 2003.

[2] J R Cary, J Candy, R H Cohen, S Krasheninnikov, D C McCune, D J Estep, J Larson, A D Malony, A. Pankin et al. 2008. First results from core-edge parallel composition in the FACETS project *J. Physics: Conf. Series* **125** 012040.

[3] John R. Cary, Ammar Hakim, Mahmood Miah, Scott Kruger, Alexander Pletzer, Svetlana Shasharina, Srinath Vadlamani, Alexei Pankin, Ronald Cohen, Tom Epperly, Tom Rognlien, Richard Groebner, Satish Balay, Lois McInnes, and Hong Zhang, "FACETS—A Framework for Parallel Coupling of Fusion Components," *The 18th Euromicro International Conference on Parallel, Distributed and Network-Based Computing*. Pisa, Italy. 2010.

[4] J. Kinsey, G. Staebler, and R. Waltz, "Burning plasma confinement projections and renormalization of the GLF23 drift-wave transport model," *Fus. Sci. Tech.*, vol. 44, p. 763, 2003.

[5] R. Groebner, A. Leonard, P. Snyder, T. Osborne, C. Maggi, M. Fenstermacher, C. Petty, and L. Owen, "Progress towards a predictive model for pedestal height in diii-d," *Nucl. Fusion*, vol. 49, p. 085037, July 2009.

[6] J. E Kinsey, G. M Staebler, and R. E Waltz, "Predicting core and edge transport barriers in tokamaks using the GLF23 drift-wave transport model," *Physics of Plasmas*, vol 12 (15), p. 052503, 2005.

[7] C. S. Chang and F. L Hinton, "Effect of impurity particles on the finite-aspect ratio neoclassical thermal conductivity in a tokamak," *Physics of Fluids*, vol 29 (10), pp. 3314–3316, 1986.

[8] T Rafiq, AY Pankin, G Bateman, A. H Kritz, and F. D Halpern, "Simulation of electron thermal transport in H-mode discharges," *Physics of Plasmas*, vol 16, p. 032505, 2009.

[9] Alexei Y. Pankin, Alex Pletzer, Srinath Vadlamani, John R. Cary, Ammar Hakim, Scott E. Kruger, Mahmood Miah, Thomas D. Rognlien, Svetlana Shasharina, Glenn Bateman, Arnold H. Kritz, and Tariq Rafiq. Simulation of anomalous transport in tokamaks using the FACETS code. *Computer Physics Communications*, In Press, Corrected Proof 2010. (doi:10.1016/ j.cpc.2010.07.03).

[10] T. Rognlien, J. Milovich, M. Rensink, and G. Porter, "A fully implicit, time dependent 2-d fluid code for modeling tokamak edge plasmas," *Journal of Nuclear Materials*, vol. 196–198, pp. 347–351, 1992.

[11] Thomas D Rognlien, Dmitri D Ryutov, N Mattor, and G. D Porter. Two-dimensional electric fields and drifts near the magnetic separatrix in divertor tokamaks. *Physics of Plasmas*, 6:1851, 1999.

[12] G M Staebler, J.E Kinsey, and R.E Waltz. A theory-based transport model with comprehensive physics. *Physics of Plasmas*, 14:055909, May 2007.

Gyrokinetic simulations of edge pedestal turbulence

Weigang Wan,[1] Scott E. Parker,[1] Yang Chen,[1] Gun-Young Park,[2] C-S Chang,[2] Norbert Podhorszki,[3] and Scott A. Klasky[3]

[1]Center for Integrated Plasma Studies, University of Colorado at Boulder, 390 UCB, Boulder, CO 80309
[2]New York University, New York, NY 10003
[3]Oak Ridge National Laboratory, Oak Ridge, TN 37830

E-mail: Weigang.Wan@Colorado.Edu

Abstract. First-principle studies of edge pedestal microinstabilities are performed with a coupled global gyrokinetic GEM-XGC0 simulation. The code coupling is implemented under the End-to-End Framework for Fusion Integrated Simulation (EFFIS).The equilibrium density, temperature and radial electric field (E_r) profiles are calculated by the XGC0 code and the anomalous transport coefficients are calculated by the GEM code. The turbulence from simulation results peaks at the position of strongest pressure gradient and hence the minimum E_r. The dominant linear instability is driven by the strong density gradient. As a result, the H-mode pedestal is linearly more unstable than L-mode. However, the nonlinear steady-state flux level of H-mode is indeed lower than L-mode. The linear instability requires kinetic electrons but is not due to trapped electrons, and it is nearly insensitive to ion and electron temperature gradients. Hence, the dominant instability is likely to be electron drift wave. It is also found in the pedestal that E_r nearly balances the diamagnetic pressure term in the radial force balance equation.

1. Introduction

To achieve high fusion power in a tokamak plasma such as the International Thermonuclear Experimental Reactor (ITER) [1], an H-mode (high-confinement mode) edge pedestal plasma is crucial. The physical mechanism of the transition from an L-(low confinement-) mode to H-mode, as well as the transport and turbulent properties of both modes are still not well understood. It is desirable, therefore, to have first-principle simulations that self-consistently evolve these physical processes. However, such simulations are difficult because multiple spatial and time scales are involved. The current state of the art is for different simulation codes to model specific physical scales. The focus of the Center for Plasma Edge Simulation (CPES), a SciDAC (Scientific Discovery through Advanced Computing) project, is to study the edge plasma with the interplay of physics from different scales, including neoclassical transport, microturbulence, and magnetohydrodynamic (MHD) instabilities. One approach is to couple several codes together for long time simulation of edge pedestal evolution. For example, four codes, including XGC0, M3D-IMP, ELITE, and M3D-MPP are coupled to study the MHD instability known as edge localized modes (ELMs) [2,3].

Here we present results of simulations of the edge pedestal plasma with code coupling of GEM and XGC0. GEM [4] is a gyrokinetic δf particle code that is suitable for the short turbulence scale but lacks the capability to self-consistently evolve the plasma profiles and equilibrium radial electric field

(E_r). XGC0 [5], on the other hand, is a full-*f* particle code suitable for the longer transport time scale provided that the transport coefficients are given. With the two codes coupled together, GEM can calculates the electromagnetic turbulent transport coefficients, and XGC0 can evolve the equilibrium profiles.

The long term goal is to use this scheme to study the full process of the L-H transition. Currently, with the coupled simulation we have studied properties of L-mode and H-mode pedestal turbulence. We find that the main instability is driven by the density gradient, a phenomenon very different from the ion-temperature gradient (ITG) driven turbulence usually observed at core plasmas. The H-mode plasma is linearly more unstable than L-mode, and it is a nonlinear effect that the H-mode has lower transport levels. Furthermore, the shape of E_r, which is believed to be a main stabilizing effect, is found to balance the equilibrium pressure profile.

2. GEM-XGC0 Code Coupling

The code coupling scheme is implemented under the EFFIS (End-to-End Framework for Fusion Integrated Simulation) framework [6,7,8], specifically developed by computer scientists to facilitate the integrated plasma edge study as part of CPES. The EFFIS framework consists of three components. One is the ADIOS (ADaptable I/O System), which is a binary data format and it provides portable, fast, scalable and easy-to-use metadata rich output with a simple API. Another main part of EFFIS is the Kepler workflow system that can orchestrate the coupling of several parallel codes running on different platforms simultaneously. Finally, the relevant physical quantities from each code at any time throughout the simulation can be selected and displayed via the Dashboard, a monitoring tool implemented using AJAX and flash. A scientist can directly view the data simply by using a standard web browser.

The workflow scheme of the GEM-XGC0 code coupling is shown in Figure 1. The workflow program runs on the Infiniband cluster *ewok*, the XGC0 code is run on the Cray XT4 supercomputer *jaguar*, and the GEM code is run on another Cray XT4 supercomputer *franklin*. Initially, the workflow starts the XGC0 code, which starts the simulation with a set of DIII-D [9] experimental profiles and typical anomalous transport parameters, and the simulation should stop before the profiles change too much (e.g., less than 1%). The new profiles calculated by XGC0, including electron and ion density, temperatures and E_r, are transfered to the GEM code via the workflow. In order to reduce numerical noise, the density and temperature profiles are fitted into some modified hyperbolic tangent functions defined by five parameters each, similar to Ref. [10]. The GEM code then runs with these profiles until a steady state is reached, and the particle and heat transport coefficients are transferred back to XGC0 for a new cycle. Throughout the process the simulation results of both codes can be viewed in real time from the Dashboard, as shown in Figure 2.

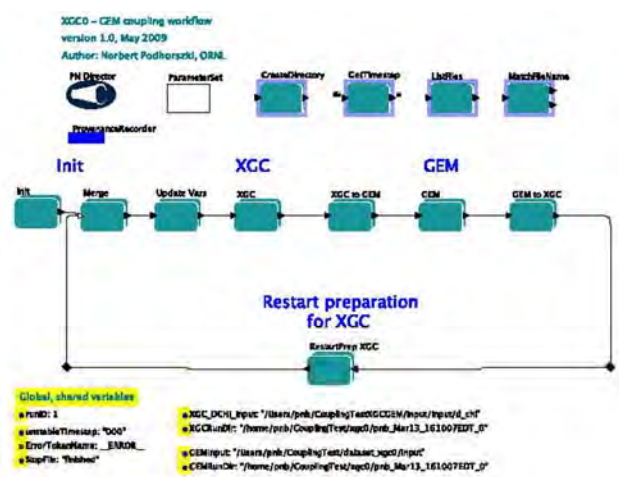


Figure 1. The GEM-XGC0 coupling workflow.

One limitation to the coupling scheme is that XGC0's magnetic geometry includes the separatrix and the X-point, while the GEM simulation region is inside the separatrix. It is then necessary for GEM to smooth the profiles from XGC0 at the radial boundaries [11]. The smoothing causes the transport coefficients calculated by GEM to be artificially small near the boundary. One way to overcome this limitation is only pass the transport coefficients at center of the simulation box back to XGC0, as uniform parameters. This may be reasonable when the simulation region is small. The other

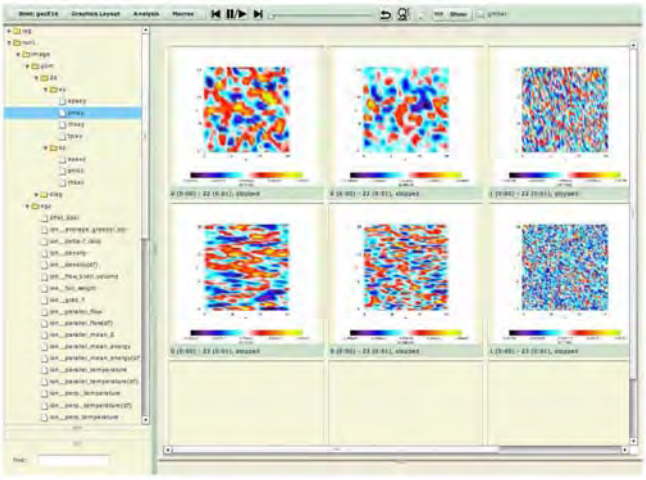


Figure 2. A screen shot of the dashboard.

approach is to extend the profiles in GEM outside of the separatrix. Both methods are implemented and tested, nevertheless in the work reported here only the first approach is used.

3. Physical Results

In this section we report preliminary results from the coupled simulation, specifically the GEM simulation of the profiles provided by XGC0. In studying these profiles we begin to obtain basic properties of L and H-mode edge plasmas. The temperature, density and radial electric field profiles from XGC0 are shown in Figure 3. In generating these profiles, XGC0 has used preset anomalous diffusion coefficients. For the L-mode, the ion particle transport coefficient $D_i = 0.5\ m^2/s$, and the heat diffusivity $\chi_i = 2.0\ m^2 s$. For the H-mode, much lower coefficients of $D_i = 0.05\ m^2/s$ and $\chi_i = 0.1\ m^2 s$ are applied for the steep pedestal, and the same D_i and χ_i as the L-mode are applied on the top. Compared to L-mode, the width of the H-mode pedestal is smaller and the height is bigger. The characteristics of these simulated profiles are consistent with experimental observations [10,12].

One important discovery from these profiles is that the radialelectric field and the equilibrium diamagnetic pressure are balanced. Theoretically, the radial electric field

$$E_r = \frac{1}{en_i}\frac{\partial}{\partial r}P_i + (\boldsymbol{u} \times \boldsymbol{B})_r. \tag{1}$$

We compare the XGC0 simulated E_r with the pressure gradient term of Eq. (1) in Figure 4, and they almost match. This means the flow contribution is nearly negligible. Since E_r is usually difficult to measure in experiments, this may enable researchers to calculate E_r directly from the pressure profile. The peak of E_r appears at the maximum of the pressure gradient, and because the shear of $\boldsymbol{E} \times \boldsymbol{B}$ flow has a stabilizing effect (see, e.g., Ref. [11]), this implies that turbulent instability should also peak there with a zero E_r shear, and the phenomenon has been confirmed by the mode structures in GEM simulations.

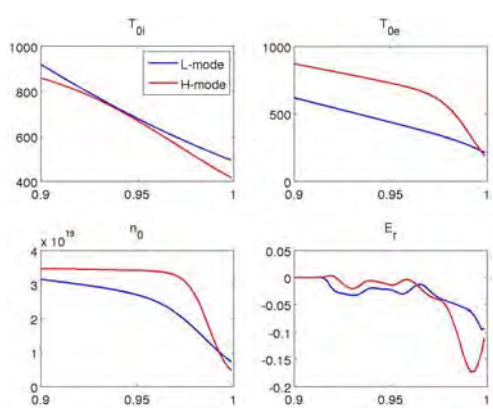


Figure 3. The profiles from XGC0.

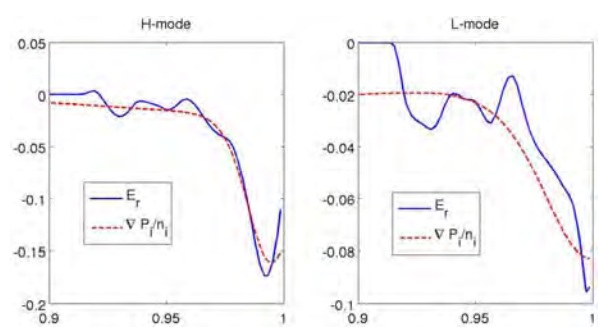


Figure 4. E_r compared to ∇P.

The nonlinear evolution of the ion particle and heat fluxes in GEM simulation are shown in Figure 5. During the initial linear growth stage, it appears that the H-mode is more unstable than the L-mode. In fact, the linear growth rate of the L-mode pedestal is $\gamma = 7.2 \times 10^{-4}\ \Omega_i$, where Ω_i is the ion cyclone frequency, while for H-mode $\gamma = 8.1 \times 10^{-4}\ \Omega_i$. It is only at the nonlinear steady state that the H-mode case has lower transport level. More investigation is needed to understand the stabilizing mechanisms in H-mode. It may be bigger E_r or zonal flows that have a stronger steering effect on fully developed turbulent eddies.

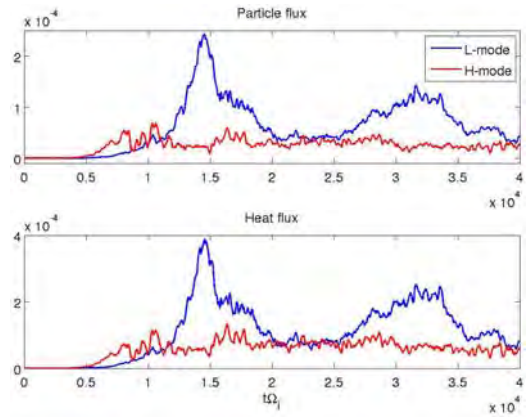


Figure 5. The volume-averaged ion particle and heat fluxes.

A detailed GEM simulation study of the linear instability shows that the most unstable mode moves in the opposite direction of ITG, meaning it is an electron mode. Indeed, the dominant instability is stabilized without kinetic electrons, and it seems to be caused by passing electrons since when the growth rate increases slightly without trapped electrons. The instability is insensitive to ion temperature gradient and is only weakly dependent on electron temperature gradient. In general, the instability is mostly dependent on density gradient, with a bigger growth rate for bigger ∇n, and this explains why H-mode is linearly more unstable. These properties lead to the conclusion that the dominant instability is electron drift wave.

It has been proposed recently [13] that the pedestal width is limited by the onset of kinetic ballooning mode (KBM) near the maximum of pressure gradient at the edge. In a set of simulations with fixed pedestal height but varying pedestal width, we find that the linear growth rate is first lowered then increased as the pedestal width get thinner and thinner, a phenomenon consistent with KBM. The authors are working on further studies for the evidence of KBM at the pedestal.

Acknowledgments
This work is supported by the DOE SciDAC projects Gyrokinetic Plasma Simulation Center and Center for Plasma Edge Simulation.

References

[1] R. Aymar et al. Overview of ITER-FEAT—The future international burning plasma experiment. *Nucl. Fusion*, 41:1301, 2001.
[2] G. Park, J. Cummings, C. S. Chang, N. Podhorszki, S. Klasky, S. Ku, A. Pankin, R. Samtaney, A. Shoshani, P. Snyder, H. Strauss, L. Sugiyama, and the CPES Team. Coupled simulation of kinetic pedestal growth and MHD ELM crash. *J. Phys.: Conf. Ser.*, 78:012087, 2007.
[3] J. Cummings, A. Pankin, N. Podhorszki, G. Park, S. Ku, R. Barreto, S. Klasky, C. S. Chang, H. Strauss, L. Sugiyama, P. Snyder, D. Pearlstein, B. Ludascher, G. Bateman, A. Kritz, and the CPES Team. Plasma edge kinetic-mhd modeling in tokamaks using kepler workflow for code coupling, data management and visualization. *Commun. Comput. Phys.*, 4(3):675–702, 2008.
[4] Y. Chen and S. E. Parker. Electromagnetic gyrokinetic δf particle-in-cell turbulence simulation with realistic equilibrium profiles and geometry. *J. Comput. Phys.*, 220:839, 2007.
[5] C. S. Chang, S. Ku, and H. Weitzner. Numerical study of neoclassical plasma pedestal in a tokamak geometry. *Phys. Plasmas*, 11:2649, 2004.
[6] Norbert Podhorszki, Bertram Ludaescher, and Scott A. Klasky. Workflow automation for processing plasma fusion simulation data. *Proceedings of the 2nd workshop on Workflows in support of large-scale science*, June 25:35–44, 2007.

[7] S. Klasky, R. Barreto, A. Kahn, M. Parashar, N. Podhorszki, S. Parker, D. Silver, and M. A. Vouk. Collaborative visualization spaces for petascale simulations. *International Symposium on Collaborative Technologies and Systems*, May 19–23:203–211, 2008.

[8] Julian Cummings, Jay Lofstead, Karsten Schwan, Alexander Sim, Arie Shoshani, Ciprian Docan, Manish Parashar, Scott Klasky, Norbert Podhorszki, and Roselyne Barreto. EFFIS: An end-to-end framework for fusion integrated simulation. *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pdp:428–434, 2010.

[9] J. L. Luxon. A design retrospective of the DIII-D tokamak. *Nucl. Fusion*, 42:614, 2002.

[10] R. J. Groebner and T. N. Carlstrom. Critical edge parameters for H-mode transition in DIII-D. *Plasma Phys. Control. Fusion*, 40:673, 1998.

[11] Yang Chen, Scott E. Parker, Gregory Rewoldt, Seung-Hoe Ku, Gun-Young Park, and C-S Chang. Coarse-graining the electron distribution in turbulence simulations of tokamak plasmas. *Phys. Plasmas*, 15:055905, 2008.

[12] T. N. Carlstrom, K. H. Burrell, R. J. Groebner, A. W. Leonard, T. H. Osborne, and D. M. Thomas. Comparison of L-H transition measurements with physics models. *Nucl. Fusion*, 39:1941, 1999.

[13] P. B. Snyder, R. J. Groebner, A. W. Leonard, T. H. Osborne, and H. R. Wilson. Development and validation of a predictive model for the pedestal height. *Phys. Plasmas*, 16:056118, 2009.

Non-Diffusive Momentum Transport and Intrinsic Rotation in Tokamaks

W. X. Wang,[1] S. Ethier,[1] S. Klasky,[2] M. Adams,[3] T. S. Hahm,[1] G. Rewoldt,[1] W. W. Lee,[1] W. M. Tang,[1] and P. H. Diamond[4]

[1]Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, New Jersey 08543
[2]Oak Ridge National Laboratory
[3]Columbia University
[4]University of California, San Diego

Email: wwang@pppl.gov

Abstract. This paper focuses on using state-of-art gyrokinetic simulations which exploit high performance computational resources to understand toroidal momentum transport and flow generation, which are complex transport phenomena of great importance in magnetic confinement fusion. Critical issues addressed are closely coupled to experimental and theoretical studies with emphasis on electron transport dominated regimes. The key results of our simulations include the identification of nonlinear flow generation by the residual stress (a non-diffusive element of the momentum flux) produced by electrostatic turbulence of ion temperature gradient (ITG) modes and trapped electron modes (TEM). This residual stress is shown to drive intrinsic rotation as a type of wave driven flow phenomenon which operates via wave-particle momentum exchange. The characteristic dependence of non-diffusive momentum transport has been studied over a wide range of experimentally-relevant parameters for the purpose of developing predictive capability for plasma rotation in ITER. Specifically, the "intrinsic" torque associated with residual stress is shown to increase close to linearly with the plasma pressure gradient and the inverse of the plasma current in various turbulence regimes. These results are consistent with experimental trends observed in various toroidal devices. Highlighted results also include robust radial pinches driven by CTEM turbulence, which are shown to play remarkable roles in determining plasma transport. Particularly, the "flow pinch" result from simulations amazingly reproduces the experimental phenomenon of radially inward penetration of perturbed flows created by modulated beams in peripheral regions, and thus is highly illuminating. Also discussed are experimental tests proposed to validate simulation predictions.

1. Introduction

Momentum transport and plasma flow generation are complex transport phenomena of great importance in magnetic confinement fusion. An optimized plasma flow is believed to play a critical role in both controlling macroscopic plasma stability, and in reducing energy loss due to plasma microturbulence. On the other hand, toroidal momentum transport is observed to be highly anomalous, non-diffusive and non-local in nearly all machines. A striking phenomenon found in experiments is the intrinsic or spontaneous rotation; namely, toroidal plasmas can self-organize and develop rotation without an external torque [1]. In fact, the intrinsic rotation in fusion plasmas is an example of a "negative viscosity phenomenon" in which an up-gradient component of the momentum flux organizes

a structured mean flow. Negative viscosity phenomena are of broad interest in the context of atmospheres, oceans, stellar interiors, and other rotating fluids. In current fusion experiments, a large plasma rotation can be driven by neutral beam injection which also provides momentum input while heating the plasma. In large size burning plasmas, however, the use of neutral beams for plasma heating becomes very challenging. It is expected that intrinsic rotation will dominate in future burning plasma experiments. Therefore, understanding the non-diffusive momentum transport mechanisms and the intrinsic rotation phenomenon is a key to predicting plasma flow in ITER.

Recently, extensive experimental studies have been carried out on this topic. The parametric dependence of the intrinsic rotation has been statistically characterized using a broad range of experimental data bases obtained in multiple machines. Specifically, the increment of central intrinsic rotation is shown to increase with the increment of plasma stored energy and to scale with the inverse of the plasma current [2] (the so called Rice scaling). Similar empirical scaling is also observed in other devices including JT-60U [3] and LHD [4], where the intrinsic rotation velocity is shown to increase with the ion pressure gradient in core plasma with an internal transport barrier. There is no doubt that these results are important for making a qualitative projection of plasma rotation in ITER. A more fundamental, critical issue is to understand the underlying physical origins of the experimental empirical scalings. This is the major focus of this study.

Out of various possibilities of physical dynamics which may play roles in determining toroidal rotation, the strong coupling between toroidal momentum and energy transport generally observed in fusion experiments [5] suggests that micro-turbulence is a key player. For turbulence driven toroidal momentum flux, a generic structure can be expressed as follows:

$$\Gamma_\phi \propto -\chi_\phi \frac{\partial U_\phi}{\partial r} + V_p U_\phi + \Pi^{\mathrm{rs}}_{r,\phi}.$$

In addition to diffusion (first term), there are two nondiffusive components, momentum pinch (second term) and residual stress (third term). The three components in the momentum flux are highly distinctive not only formally but also physically. Besides their different physical origins under turbulence circumstances, they have qualitatively distinct effects on the toroidal flow formation. The diffusive transport is well known in the direction opposite to the rotation gradient, leading to the relaxation of the rotation profile and the release of associated free energy. The momentum pinch term is a convective flux which is directly proportional to the rotation velocity U_ϕ with V_p as the pinch velocity. Both momentum diffusion and pinch can move plasma mechanical momentum (i.e., toroidal momentum carried by particles), and then rearrange the rotation profile, radially. A qualitative distinction is that momentum pinch can transport momentum in either direction, up-gradient or down-gradient.

The residual stress $\Pi^{\mathrm{rs}}_{r,\phi}$, which is defined as a specific part of the Reynolds stress with no direct dependence on either the rotation velocity or its gradient, can be shown in the momentum transport equation to be isomorphic in mathematical form to the integrated external momentum source which acts as a torque to drive the rotation. Thus, the residual stress can act as an internal local torque to spin up a plasma, offering an ideal mechanism to drive intrinsic rotation under appropriate boundary conditions. For this reason, the quantity $\nabla\cdot\mathbf{\Pi}^{\mathrm{rs}}_{r,\varphi}$ is widely referred to as the intrinsic torque in experimental and theoretical investigations. In a broad physical context, this is a type of wave-driven flow phenomenon which operates via wave-particle momentum exchange [6]. Note that all three components have been observed in tokamak experiments. Searching for nondiffusive elements and understanding underlying mechanisms have been the focus of recent intensive theoretical and experimental efforts.

In this paper, new results of non-diffusive toroidal momentum transport found from our global gyrokinetic simulations are reported. We focus our study on understanding the nonlinear residual stress generation and its effect on toroidal flow formation in electrostatic turbulence regimes of ion

temperature gradient (ITG) modes and trapped electron modes (TEM). This study concerns a few critical issues which are highly relevant to experimental observations and theoretical studies.

2. Gyrokinetic Simulation Models of Rotation Plasma and Treatment of Kinetic Electrons

In this work, turbulence-driven non-diffusive momentum transport is investigated using the global Gyrokinetic Tokamak Simulation (GTS) code [7] with emphasis on the characteristic dependence of turbulence driven intrinsic rotation on plasma parameters. The GTS code is based on a generalized gyrokinetic simulation model using a δf particle-in-cell approach, and incorporates the comprehensive influence of non-circular cross section, realistic plasma profiles, plasma rotation, neoclassical (equilibrium) electric field, Coulomb collisions, and other features. It can directly read plasma profiles of temperature, density and toroidal angular velocity from the TRANSP experimental database, and a numerical magnetohydrodynamic (MHD) equilibrium reconstructed by MHD codes using TRANSP radial profiles of the total pressure and the parallel current (or safety factor), along with the plasma boundary shape.

First, we give a brief description on our gyrokinetic simulation model for rotating plasmas in this section. In a δf simulation, the turbulence fluctuations are considered as perturbations on the top of the neoclassical equilibrium. The gyrokinetic particle distribution function is expressed as $f = f_0 + \delta f$. The equilibrium distribution function f_0 of ions, with magnetic moment μ and parallel velocity $v_{\|}$ as independent velocity variables, is determined by the neoclassical dynamics and obeys

$$\frac{\partial f_0}{\partial t} + (v_{\|}\hat{b} + \mathbf{v}_{E_0} + \mathbf{v}_d)\cdot\nabla f_0 - \hat{\mathbf{b}}^*\cdot\nabla(\mu B + \frac{e}{m_i}\Phi_0)\frac{\partial f_0}{\partial v_{\|}} = C_i(f_0, f_0). \tag{1}$$

Here, $\mathbf{v}_{E_0}$ is the magnetic drift velocity corresponding to the equilibrium potential Φ_0. $\mathbf{v}_d$ is the ∇B drift velocity, $\hat{\mathbf{b}}^* = \hat{\mathbf{b}} + \rho_{\|}\,\hat{\mathbf{b}}\times(\hat{\mathbf{b}}\cdot\nabla\hat{\mathbf{b}})$ with $\hat{\mathbf{b}} = \mathbf{B}/B$, C_i is the Coulomb collision operator, and e and m_i are the ion charge and mass, respectively. The lowest order solution of Eq. (1) is a shifted Maxwellian consistent with (large) plasma rotation [8]:

$$f_0 = f_{SM} = n(r,\theta)(\frac{m_i}{2\pi T_i})^{3/2} e^{-\frac{m_i}{T_i}[\frac{1}{2}(v_{\|}-U_i)^2+\mu B]}, \tag{2}$$

where the parallel flow velocity U_i is associated with the toroidal rotation by $U_i = I\omega_\phi/B$ with ω_ϕ the toroidal angular velocity and I the toroidal current, and $n_i(r, \theta)$ is the ion density, $n_i(r,\theta) = N(r)e^{\frac{m_i U_i^2}{2T_i} - \frac{e\tilde{\Phi}_0}{T_i}}$, with poloidal variation associated with plasma rotation. The total equilibrium potential consists of two parts, $\Phi_0 = \langle\Phi_0\rangle + \tilde{\Phi}_0$. Here, $\langle\ \rangle$ denotes a flux surface average. The poloidally varying component $\tilde{\Phi}_0$ can be generated by the centrifugal force which drives charge separation on a magnetic surface in strongly rotating plasmas [9]. Generally the radial potential $\langle\Phi_0\rangle$ is dominant. The equilibrium radial electric field can be calculated from a first-principles based particle simulation of neoclassical dynamics with important finite orbit effects [8], or obtained by direct experimental measurement if available. Instead of using a true neoclassical equilibrium distribution function, which is unknown analytically, we use this lowest order solution for equilibrium toroidal plasmas in the present simulations. A shifted Maxwellian with either model or experimental profiles of $\langle n(r,\theta)\rangle$, $T_i(r)$ and $\omega_\phi(r)$ is prescribed for the ions. In the electrostatic limit, the ion gyrokinetic equation for the turbulence perturbed distribution δf_i of ion guiding centers is

$$\frac{\partial \delta f_i}{\partial t} + (v_\parallel \hat{b} + \mathbf{v}_{E_0} + \mathbf{v}_E + \mathbf{v}_d) \cdot \nabla \delta f_i - \hat{\mathbf{b}^*} \cdot \nabla(\mu B + \frac{e}{m_i}\Phi_0 + \frac{e}{m_i}\bar{\Phi})\frac{\partial \delta f_i}{\partial v_\parallel} =$$

$$\left\{ -\left[\frac{m}{T_i}\left(\frac{1}{2}(v_\parallel - U_i)^2 + \mu B\right) - \frac{3}{2}\right] \mathbf{v}_E \cdot \nabla \ln T - \mathbf{v}_E \cdot \nabla \ln n(r,\theta) - \frac{m(v_\parallel - U_i)}{T_i}\mathbf{v}_E \cdot \nabla U_i(r,\theta) \right.$$

$$\left. + \frac{mU_i}{T_i v_\parallel}\mathbf{v}_E \cdot \mu \nabla B - \frac{1}{T_i}(v_\parallel \hat{b} + \mathbf{v}_d) \cdot \nabla(e\bar{\Phi})(1 - \frac{U_i}{v_\parallel}) \right\} f_0 + C_i^l(\delta f_i). \tag{3}$$

Here, $\mathbf{v}_E$ is the $\mathbf{E} \times \mathbf{B}$ velocity corresponding to the fluctuation potential $\bar{\Phi}(\mathbf{R})$ at the ion guiding center coordinates $\mathbf{R}$, and C_i^l is the linearized Coulomb collision operator. On the right hand side, the third term proportional to ∇U_i is the Kelvin-Helmholtz-type drive term. The other terms containing U_i are also retained, which can be important when the Mach number of plasma flow is high.

The GTS code solves the gyrokinetic Poisson equation in configuration space for the turbulence potential $\Phi(\mathbf{x})$ at the particle coordinates $\mathbf{x}$. Unlike in flux-tube or wedge codes, the real space, global Poisson solver, in principle, retains all toroidal modes from (m/n =0/0) all the way to a limit which is set by grid resolution, and therefore retains full-channel nonlinear energy couplings. There are two largely different Poisson solvers implemented in the GTS simulation. In a simple geometry limit, i.e., large aspect ratio and circular cross section, turbulence fluctuations $\delta\Phi$ on small spatial and fast time scales and axisymmetric zonal flow $\langle\Phi\rangle$ on larger (meso-scale) spatial and slow time scales can be decoupled using i) a Pade approximation, i.e., $\Gamma_0(b) \equiv I_0(b)e^{-b} \approx 1/(1 + b)$ with I_0 the modified Bessel function and $b = (k_\perp\rho_i)^2$, and ii) $\langle\tilde{\Phi}\rangle \approx \langle\widetilde{\Phi}\rangle$, , i.e., operations between the flux surface average $\langle\Phi\rangle$ and the gyrokinetic double average $\tilde{\Phi}$ can commute. This results in two decoupled equations [7]:

$$\frac{e}{T_i}\left(\delta\Phi - \widetilde{\delta\Phi}\right) = \frac{\delta\bar{n}_i - \langle\delta\bar{n}_i\rangle}{n_0} - \frac{\delta n_e - \langle\delta n_e\rangle}{n_0}, \tag{4}$$

$$\frac{1}{\mathcal{V}_r'}\frac{d}{dr}\left[\frac{d\langle\Phi\rangle}{dr}\mathcal{V}_r'\langle g^{rr}\rangle\right] = \frac{1}{\mathcal{V}_r'}\frac{d}{dr}\left\{\frac{d}{dr}\left[\frac{T_i}{e}\left(\frac{\langle\delta\bar{n}_i\rangle}{n_0} - \frac{\langle\delta n_e\rangle}{n_0}\right)\right]\mathcal{V}_r'\langle g^{rr}\rangle\right\}$$
$$-\langle\frac{1}{\rho_i^2}\rangle\frac{T_i}{e}\left(\frac{\langle\delta\bar{n}_i\rangle}{n_0} - \frac{\langle\delta n_e\rangle}{n_0}\right), \tag{5}$$

where $\delta\bar{n}_i(\mathbf{x})$ and $\delta n_e(\mathbf{x})$ are the ion and electron density fluctuations, respectively, $\mathcal{V}_r' \equiv d\mathcal{V}/dr$, with $\mathcal{V}$ the volume enclosed by magnetic surface r, and $g^{rr} \equiv \nabla r \cdot \nabla r$. Because turbulence dynamics on different spatio-temporal scales are separated in solving the Poisson equation, the advantages are apparent. However, the above approximations, particular the second one, are not well justified in general toroidal geometry. This has motivated us to develop a generalized Poisson solver which solves an integral equation for the total potential $\Phi = \delta\Phi + \langle\Phi\rangle$,

$$\frac{e}{T_i}\left(\Phi - \widetilde{\Phi}\right) = \frac{\delta\bar{n}_i}{n_0} - \frac{\delta n_e}{n_0}. \tag{6}$$

While the adiabatic electron model has been widely used for simplicity in many earlier numerical and theoretical studies of ITG driven turbulence, non-adiabatic electron physics is in general irreducible in turbulence dynamics of toroidal systems. For ITG and TEM turbulence with $k_\perp\rho_e \ll 1$, we use a drift kinetic description for electrons, neglecting the finite gyroradius effect. However, for

electron gyroradius scale turbulence, such as electron temperature gradient driven turbulence, electrons are treated as fully gyrokinetic. Similarly as for ions, the δf method can be used to solve for the total perturbed electron guiding center distribution function, $\delta f_e = f_e - f_{e0}$, corresponding to turbulence fluctuations. The equilibrium distribution f_{e0} satisfies the electron version of Eq. (1) and can be approximated by a shifted Maxwellian containing a parallel flow similar to that for the ions. Apparently, δf_e contains both adiabatic and non-adiabatic electron response. Another simulation model to treat kinetic electrons is to separate the non-adiabatic electron response δh_e using $f_e = f_{e0} + (e\delta\Phi/T_e)\, f_{e0} + \delta h_e$, and to solve for the non-adiabatic part δh_e according to Eq. (16) of Ref. 7. In this case, Eq. (4) becomes

$$\left(1+\frac{T_i}{T_e}\right)\frac{e\delta\Phi}{T_i} - \frac{e\widetilde{\delta\Phi}}{T_i} = \frac{\delta\bar{n}_i - \langle\delta\bar{n}_i\rangle}{n_0} - \frac{\delta n_e^h - \langle\delta n_e^h\rangle}{n_0}, \tag{7}$$

and Eq. (6) becomes

$$\left(1+\frac{T_i}{T_e}\right)\frac{e\Phi}{T_i} - \frac{e\widetilde{\Phi}}{T_i} - \frac{e\langle\Phi\rangle}{T_e} = \frac{\delta\bar{n}_i}{n_0} - \frac{\delta n_e^h}{n_0}, \tag{8}$$

where $\delta n_e^h \equiv \int d^3v\,\delta h_e$ is the non-adiabatic electron density fluctuation.

The left-hand side of the δh_e equation (Eq. (16) of Ref. 7) contains a time derivative term, $\frac{\partial\delta\Phi}{\partial t}$, which can easily give rise to numerical instability if it is calculated using direct finite differences. To avoid the numerical problem, a split-weight scheme [10] was proposed, which uses a separate equation for calculating $\frac{\partial\Phi}{\partial t}$. The $\frac{\partial\Phi}{\partial t}$ equation, which is not a new equation, is obtained by taking the time derivative of the gyrokinetic Poisson equation and using the ion and electron continuity equations. In a toroidal system, the equation for calculating $\frac{\partial\Phi}{\partial t}$ is obtained as follows:

$$\frac{e}{T_i}\left(\frac{\partial\Phi}{\partial t} - \widetilde{\frac{\partial\Phi}{\partial t}}\right) = \frac{1}{n_0}(\nabla\cdot\mathbf{\Gamma}_e - \nabla\cdot\mathbf{\Gamma}_i) + \frac{c\mathbf{B}\times\nabla\delta\Phi}{B^2}\cdot\left(2\frac{\nabla B}{B} - \frac{e}{T_e}\nabla\Phi_0\right), \tag{9}$$

where wide-tilde denotes the gyrokinetic double average over $\frac{\partial\Phi}{\partial t}$, electron particle flux $\mathbf{\Gamma}_e \equiv \int d^3v(v_\parallel\mathbf{b} + \mathbf{v}_{E_0} + \mathbf{v}_E + \mathbf{v}_d)\delta h_e$ and ion particle flux $\mathbf{\Gamma}_i \equiv (1/2\pi)\int d^3v d\mathbf{R} d\Theta\, (v_\parallel\mathbf{b} + \mathbf{v}_{E_0} + \mathbf{v}_E + \mathbf{v}_d)\delta f_i \delta(\mathbf{R} - \mathbf{x} + \rho)$, with ρ the gyroradius vector and Θ the gyrophase. Then $\frac{\partial\delta\Phi}{\partial t}$ is calculated directly using $\frac{\partial\delta\Phi}{\partial t} \equiv \frac{\partial\Phi}{\partial t} - \left\langle\frac{\partial\Phi}{\partial t}\right\rangle$.

It is noticed that many previous simulations include only trapped electrons for the non-adiabatic electron response. Numerically, the fast parallel streaming of passing electrons gives rise to a strict constraint on the time step size, which adds to the computational challenge. While the trapped electrons are the primary origin of non-adiabatic response, some passing electrons can be non-adiabatic too. In fact, dynamical division between trapped and passing electrons is, though not impossible, highly non-trivial during simulations because of the dependence of the trapping-passing boundary on the electric potential which evolves in time, and of the collisional trapping-detrapping process. Nevertheless, thanks to the availability of supercomputing capabilities, we retain full electron dynamics by including both trapped and untrapped electrons in the simulations.

The GTS simulation has been benchmarked against other gyrokinetic codes in the electrostatic regime and the large aspect ratio circular concentric geometry limit [7, 11]. Further verification studies have been carried beyond linear benchmarks. We here present a few examples.

Unlike ITG turbulence with adiabatic electrons, non-adiabatic electron dynamics can drive particle transport for turbulence such as TEM. It is well known that turbulence driven particle transport across the magnetic field lines is ambipolar, i.e., flux-surface-averaged radial particle fluxes for electrons and ions are equal, so as to maintain the overall quasineutrality in a toroidal system. The ambipolarity property of TEM driven cross-field particle transport is tested in the GTS simulation. The time history of particle fluxes (at $r/a = 0.54$) is plotted in Fig. 1, showing that electron and ion fluxes very closely track with each other all the time during the simulation. Moreover, the ambipolarity of turbulence driven particle transport is obtained locally over the entire radial domain ($0.1 \leq r \leq 0.9$) of the global simulation, as is seen in the right panel of Fig. 1 which plots the steady state particle fluxes versus minor radius. This guarantees that quasineutrality is satisfied radially locally.

Global gyrokinetic turbulence is characterized by distinguishable dynamical phases in both coordinate space and wavenumber space [12]. Ideally, the dynamics of gyrokinetic turbulence should be robust to numerical techniques. A convergence study for CTEM turbulence is presented in Fig. 2. Two simulations using 50 and 100 particles/cell species, respectively, are shown to produce well converged results for electron particle transport that displays no noticeable difference in statistical sense (left panel). In other words, the difference in the simulated fluxes between the two cases using 50 and 100 particles/cell·species is within the same range of statistical error of different simulation runs with the same number of particles but with different initial conditions. At the mean time, the time evolution of corresponding average electron weight square, $\langle w^2 \rangle$, of the two simulations is shown to be almost identical (right panel). This result indicates that the observed weight growing does not depend on whether 50 or 100 particles/cell·species are used in these simulations, and is driven by physics, corresponding to the increase of amplitude of δf associated with plasma profile evolution induced by turbulence-driven fluxes during transport time scale. Furthermore, while the particle weight is physically growing during the simulations, there is no observable correlation between the weight evolution and the dynamics of electron particle flux, as is shown in Fig. 2. The particle weight remains at a low level ($\langle w^2 \rangle < 0.09$) at the end of the simulations, which does not impact the results of simulated transport. These convergence studies clearly indicate that the noise-induced transport in our simulations is negligible with respect to turbulence driven transport.

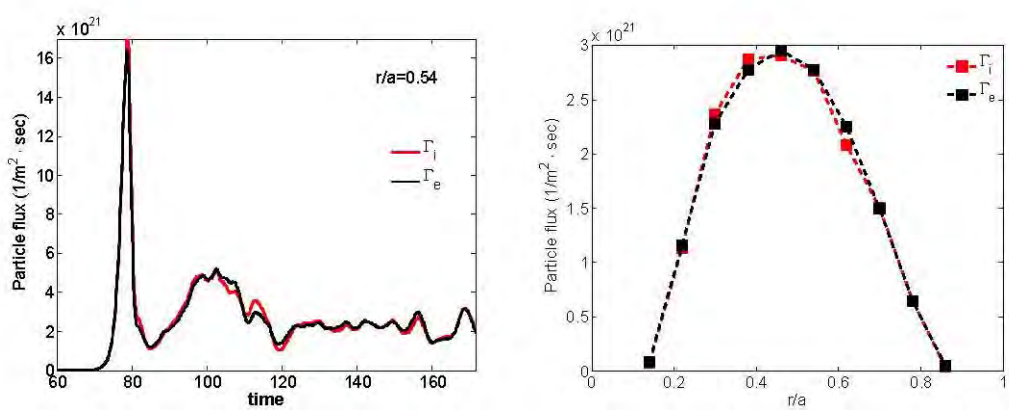


Figure 1. Time history of ion and electron particle fluxes (left), and steady state ion and electron particle fluxes versus minor radius r (right). Time unit is L_n/c_s.

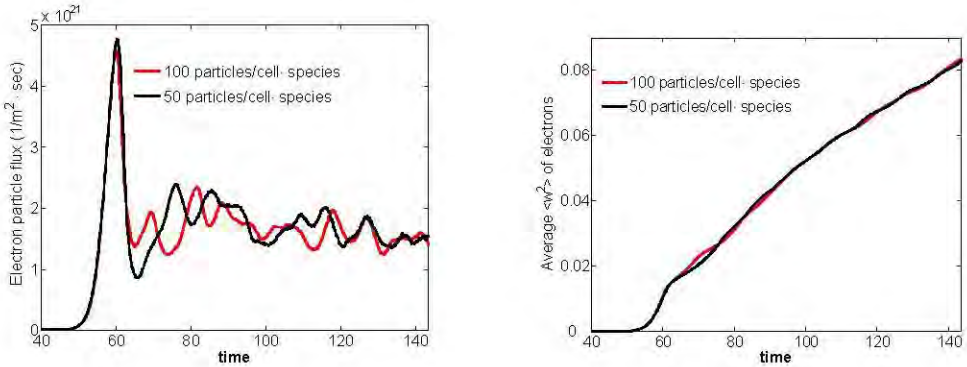


Figure 2. Time history of electron particle fluxes (left), and average weight squares of electrons (right) from two simulations using different number of simulation particle. Time unit is L_{Te}/c_s. The major parameters used are: $R_0/L_{Te} = R_0/L_n = 6.5$, $R_0/L_{Ti} = 2.4$. A shaped DIII-D type MHD equilibrium is used.

3. Nonlinear Flow Generation by Gyrokinetic Turbulence

To search for nondiffusive elements of toroidal momentum transport and understand underlying mechanisms, global gyrokinetic simulations using the GTS code have been carried out systematically over a wide range of experimentally relevant parameters. Our simulation studies have revealed an important nonlinear flow generation process due to the residual stress produced by electrostatic ITG and TEM turbulence [11, 13]. Particularly, in collisionless TEM (CTEM) turbulence, both turbulence fluctuation intensity and the intensity gradient are identified as nonlinear mechanisms for driving residual stress. Another key ingredient for turbulence driven residual stress is symmetry breaking in the parallel wave number spectrum. Such a well know mechanism is the mean $\mathbf{E} \times \mathbf{B}$ flow shear [14, 15]. For most drift wave instabilities, both signs of $k_{||}$ are equally excited in the ideal case, resulting in a reflection symmetry in the $k_{||}$ spectrum. Perfect local $k_{||}$ symmetry means perfectly balanced population density between co-and counter-propagating acoustic waves along the torus, and thus a vanishing net local momentum torque. Therefore, a critical, generic piece of physics behind the residual stress spinning up the plasma is the breaking of the $k_{||} \rightarrow -k_{||}$ symmetry. Concerning the origin of the symmetry breaking, turbulence self-generated low frequency zonal flow shear has been identified to be a key, universal mechanism in various turbulence regimes. Simulations also indicate the existence of other mechanisms beyond $\mathbf{E} \times \mathbf{B}$ shear.

Figure 3 illustrates how the residual stress is nonlinearly driven by turbulence intensity, acting with turbulence-generated zonal flow induced $k_{||}$ symmetry breaking. These results are obtained from a simulation of CTEM turbulence, using typical parameters of DIII-D plasmas. The major parameters used here are: $R_0/L_{Te} = R_0/L_n = 6.0$, $R_0/L_{Ti} = 2.4$, $T_e = 4.8$ Kev and $T_i = 3.5$ Kev at $r/a = 0.5$, and an initial rotation $\omega_\phi = 0$. First, the CTEM-driven toroidal flux exhibits coherent spatio-temporal bursting behavior with momentum flux pulses propagating both inward and outward in the radial direction, as shown in the upper-left panel of Fig. 3. Because of the zero initial toroidal rotation used, the momentum flux is, by definition, essentially residual stress. Interestingly, the residual stress at steady state changes direction from outward in the inner core region to inward in the outer core region. Plotted in the lower-left panel is $\sum k_{||}\delta\Phi_{mn}^2$, where $\delta\Phi_{mn}$ is a mode amplitude, with m and n the poloidal and toroidal mode numbers, respectively. This quantity can be used to measure the component of the residual stress driven by the turbulence intensity in the presence of non-vanishing $\langle k_{||} \rangle$. One can see that $\sum k_{||}\delta\Phi_{mn}^2$ indeed reproduces a similar spatio-temporal behavior to the directly calculated momentum flux, particularly in the inner core region ($r/a < 0.55$). In the

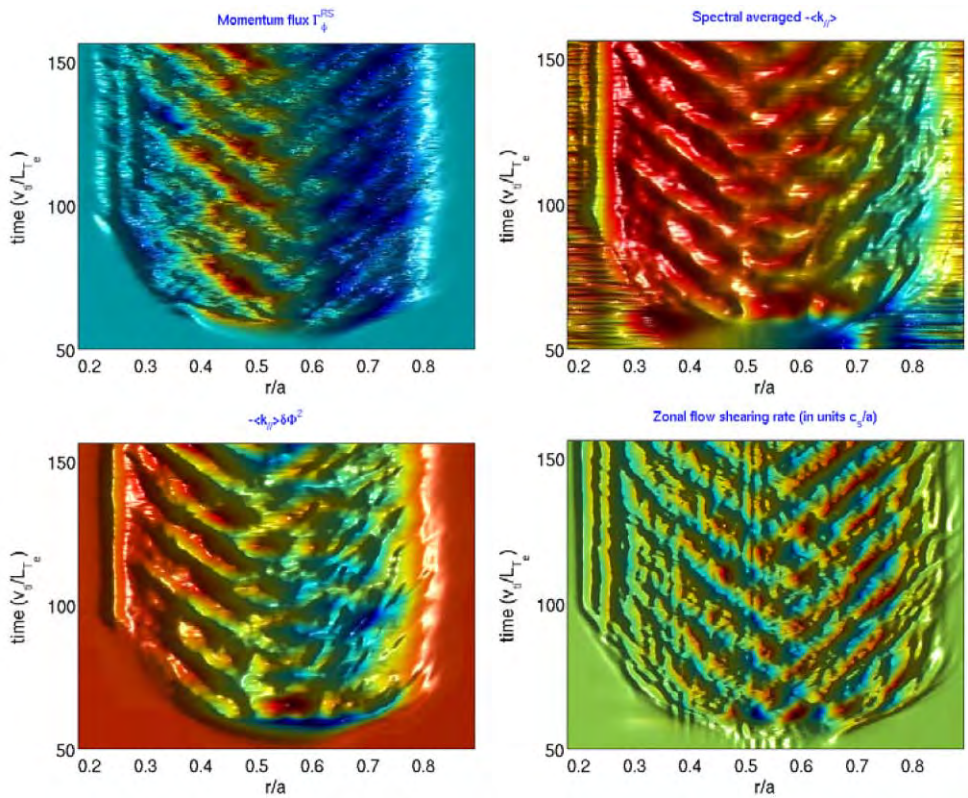


Figure 3. Spatio-temporal evolution of radial flux of toroidal momentum (upper-left), $\langle k_{\|}\rangle \sum \delta\Phi_{mn}^2$ (lower-left), spectrum-averaged $\langle k_{\|}\rangle$ (upper-right), zonal flow shearing rate (lower right).

outer core region (r/a > 0.55), the turbulence intensity gradient is identified to play significant role in driving residual stress [11]. Further, in the upper-right panel, the spectrum-averaged $k_{\|}$, defined as

$$\langle k_{\|}\rangle(r) \equiv \frac{1}{qR_0}\frac{\sum(n/|n|)(nq-m)\delta\Phi_{mn}^2}{\sum \delta\Phi_{mn}^2},$$

shows an apparent spatio-temporal correlation with $\Gamma_\phi^{\mathrm{RS}}$, indicating the importance of nonvanishing $\langle k_{\|}\rangle$. The whole picture for the residual stress generation is completed by finding out what causes $k_{\|}$ symmetry breaking, giving rise to the non-zero $\langle k_{\|}\rangle$. This is in the lower-right panel which plots the shearing rate of turbulence self-generated zonal flows according to the following expression

$$\omega_E^{\mathrm{ZF}} = \frac{R^2 B_p^2}{B}\frac{\partial}{\partial \Psi_p}\left(\frac{E^{\mathrm{ZF}}}{RB_p}\right),$$

where B and B_p are the total and poloidal magnetic field strengths, and Ψ_p is the poloidal magnetic flux. A clear correlation between the zonal flow shearing rate ω_E^{ZF} and $\langle k_{\|}\rangle$ indicates that the breaking of $k_{\|}$ symmetry and the yielding of nonvanishing $\langle k_{\|}\rangle$ are caused by the zonal flow shear. Since zonal flows are turbulence self-generated, this process represents a universal, nonlinear mechanism for residual stress generation.

4. Characteristics of Turbulence-Nonlinearly-Driven Plasma Flow and Origin of Empirical Scalings of Intrinsic Rotation

The turbulence-nonlinearly-driven residual stress, acting as an intrinsic torque, is shown to spin up toroidal rotation effectively. In our previous study, ITG turbulence driven "intrinsic" torque is shown to increase close to linearly with plasma pressure gradient [11], in qualitative agreement with experimental observations in various devices. For typical plasma parameters of fusion experiments, collisionless TEM turbulence can be a major source to drive multiple-channel transport, including toroidal momentum transport. However, the momentum transport and flow generation phenomena have not been well explored experimentally in the electron transport dominated regimes. Quantifying the characteristic dependence of turbulence generated toroidal flow in the electron turbulence regimes is particularly important for ITER experiments in which the electron channel is expected to dominate plasma transport.

The characteristic dependence of intrinsic torque driven by CTEM turbulence is numerically investigated. The GTS simulations are carried out over a wide range of experimentally relevant plasma parameters, which cover various regimes with respect to different sources of free energy for driving CTEM turbulence.

First, we examine the dependence of intrinsic rotation on electron pressure gradient. For this parametric scan study, radial profiles of electron density/temperature/pressure gradient used in simulations are specified according to the expression: $R_0/L_{n_e,T_e,p_e} = -\kappa \exp\left[-\left(\frac{\rho-\rho_c}{0.28}\right)^6\right]$, along with a fixed density/temperature/pressure at the center ρ_c = 0.5. This gives a fairly uniform CTEM drive in a region centered at ρ_c and near zero gradient elsewhere. The simulation scan is performed by varying the κ value. For all these simulations, plasmas are initially rotation-free and momentum-source-free, which allows us to concentrate on the residual stress associated intrinsic torque. An equilibrium $\mathbf{E} \times \mathbf{B}$ shear is also included via the radial force balance relation, which, however, is seen to be a minor player with respect to CTEM self-generated zonal flows. The numerical MHD equilibrium used in this study corresponds to a real DIII-D discharge. All simulations in this paper use 100 particle/cell·species.

Instead of calculating the local torque $\nabla \cdot \mathbf{\Pi}^{\mathrm{RS}}_{r,\phi}$, we examine the rate of toroidal momentum generation, dP_ϕ/dt, associated with the residual stress, with $P_\phi \equiv \int d^3r \left| \int d^3v m_i R v_\phi \delta f_i \right|$. Apparently, the quantity dP_ϕ/dt is a measure of the volume-integrated (or spatially averaged) torque driven by turbulence, which has better correspondence to the intrinsic torque inferred from experiments or measured central intrinsic rotation. The simulation results of total intrinsic torque dP_ϕ/dt driven by CTEM turbulence, versus the electron pressure gradient ∇p_e, are summarized in Fig. 4. The three curves in Fig. 4 correspond to three cases of free energy for driving CTEM. The dominant free energy sources are ∇n (black), ∇T_e (green) and a combination of both (red), respectively. For all three cases, the intrinsic torque associated with nonlinearly generated residual stress is found to increase close to linearly with the electron pressure gradient. In the other words, a larger central intrinsic rotation is expected to be produced in a plasma with a higher electron pressure gradient. The dominant underlying physics governing this scaling is rather straightforward, namely, both the turbulence intensity and the zonal flow shear, which are two key ingredients for driving residual stress, are increased with the strength of the CTEM drive R_0/L_{p_e}. Moreover, the observation of the black curve being above the green and red curves indicates that the free energy in the density gradient is more efficient in driving intrinsic rotation via CTEM turbulence. These results predicted from the gyrokinetic simulations suggest a strong connection between intrinsic rotation and electron parameters. It is highly interesting to test this prediction in experiments. As a good opportunity for validation study, particularly, NSTX experiments can be used as a unique platform to test the

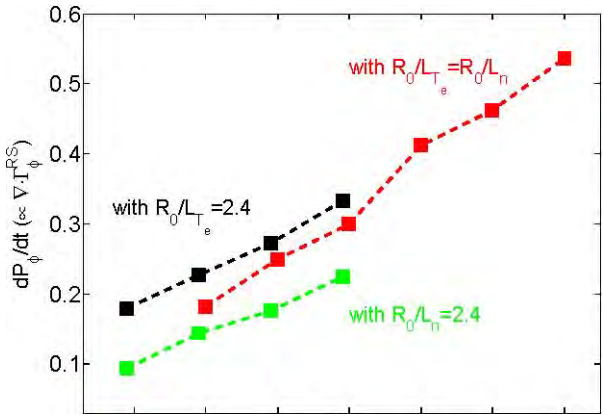


Figure 4. CTEM-driven total intrinsic torque (spatially averaged) versus electron pressure gradient R_0/L_{p_e}.

predicted characteristic dependence of intrinsic rotation on electron parameters in electron transport dominated regimes.

Now we turn to exploring the dependence of residual stress and intrinsic rotation on the plasma current I_p. Again, this simulation study is carried out for CTEM turbulence. The primary purpose is to attempt to shed light on the physics origin of the current scaling which was obtained in multiple devices [2]. For this simulation study we adopt a similar methodology to that used in experiments for various investigations of current scans. A set of simulation experiments is carried out by holding the vacuum (external) magnetic field and plasma pressure profile fixed, while varying the plasma current. Specifically, this is accomplished by generating a series of shaped, numerical equilibria with I_p = 0.75, 1.0, 1.5 and 2.0 MA, using an MHD code.

The major parameters used in this study are: R_0/L_{T_e} = R_0/L_n = 6, R_0/L_{T_i} =2.4 and 100 particles/cell·species. Simulation results presented in the right panel of Fig. 5 show that the rate of toroidal momentum generation by CTEM turbulence (i.e., total intrinsic torque) increases close to linearly with the inverse of the plasma current. This result indeed reproduces the same trend as that of the Rice scaling. With respect to the torque versus ∇T, ∇n and ∇p scaling in the ITG and CTEM turbulence, however, the underlying physics governing the current scaling is less transparent.

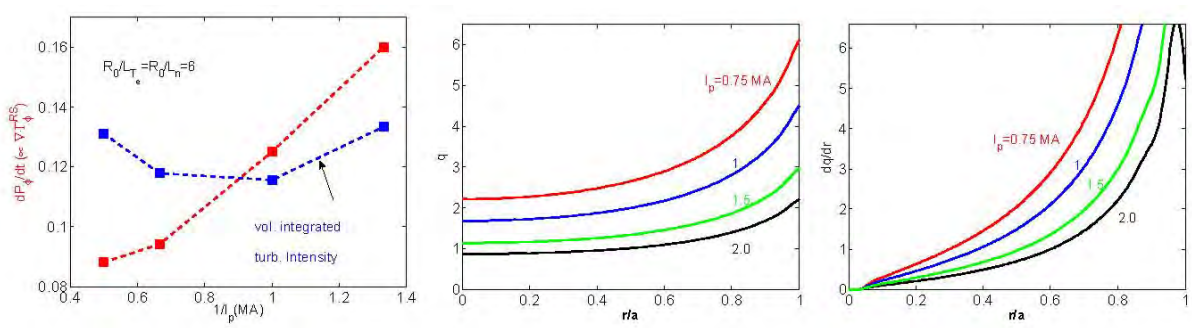


Figure 5. CTEM-driven total intrinsic torque and volume-integrated turbulence intensity at steady state versus plasma current I_p (left), and corresponding radial profiles of safety factor q (middle) and magnetic shear dq/dr (right) of the four cases.

Both turbulence intensities and intensity gradients are shown to drive the residual stress. First, we examine the turbulence intensity levels of four cases. As is also shown in the right panel of Fig. 5, the volume-integrated turbulence intensities in the steady state are actually on the same level for the four cases, roughly independent of the current. At the same time, the turbulence intensity gradient, which

can also contribute to driving residual stress with an asymmetric fluctuation spectrum in $k_{||}$ due to turbulence wave radiation induced wave momentum diffusion [16], also does not show significant current dependence that can account for the torque vs I_p scaling observed in our simulations. Hence, this implies that the underlying physics for the current scaling has to do with the symmetry breaking dynamics and the associated nonvanishing spectrally averaged $k_{||}$. As we found previously, the turbulence self-generated zonal flow shear is a key mechanism for the symmetry breaking. Simulations also indicate the existence of other mechanisms beyond $\mathbf{E} \times \mathbf{B}$ shear. These include the magnetic shear, to be discussed below.

On the other hand, the corresponding q profile is remarkably boosted in the four equilibria as the plasma current is decreased from 2 MA to 0.75 MA, as shown in the middle panel of Fig. 5. So is the magnetic shear dq/dr (left panel of Fig. 5). Note that the parameter $\rho_*(\equiv \rho_i/a)$ for the four cases is roughly the same, i.e., $a/\rho_i \sim 170$, which is in the DIII-D range. This observation is highly suggestive that the current scaling of intrinsic torque and rotation may have connections with the change in the magnetic shear and/or the value of q.

To identify the effects of the safety factor and the magnetic shear separately, further computational experiments are performed. First, we examine the effect of the q value. To this end, three MHD equilibria are created, which hold the profile of magnetic shear (and plasma pressure) fixed while boosting the q profile, as shown in the right panel of Fig. 6. As illustrated in the left panel of Fig. 6, the CTEM driven intrinsic torque is shown to decrease with the increase of the q value for this scan. The dependence of the volume-integrated turbulence intensity on q value plotted in the same figure indicates that this turbulence intensity dependence appears to be a major cause for the observed intrinsic torque vs q dependence. The key point of this interesting result, however, is that the dependence of the torque on the q value shows the opposite trend to the current scaling obtained in Fig. 5. Therefore, the current scaling is not due to the effect of the q value on the nonlinear residual stress generation.

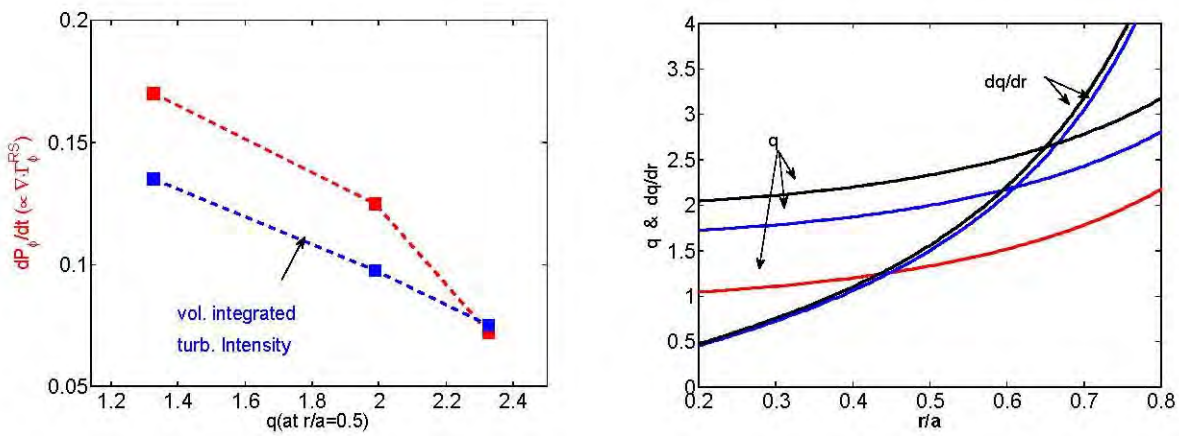


Figure 6. CTEM-driven total intrinsic torque versus q value averaged over the core region (left) and corresponding radial profiles of q and dq/dr for the three equilibria used for these simulations (right).

Now we turn to exploring the effects of the magnetic shear on the intrinsic torque. To this end, three MHD equilibria are created, which hold the radially averaged q value nearly fixed in the central core region where CTEM turbulence is generated, but allow minor variation in the q profile in order to create significant variation in the magnetic shear, as illustrated in the middle and right panels of Fig. 7. At the same time, the plasma pressure is also held fixed. Note that a normal (positive) magnetic shear is presented in these equilibria. The result of this simulation scan is presented in the left panel of Fig. 7, which shows that the volume-integrated intrinsic torque increases nearly linearly with the

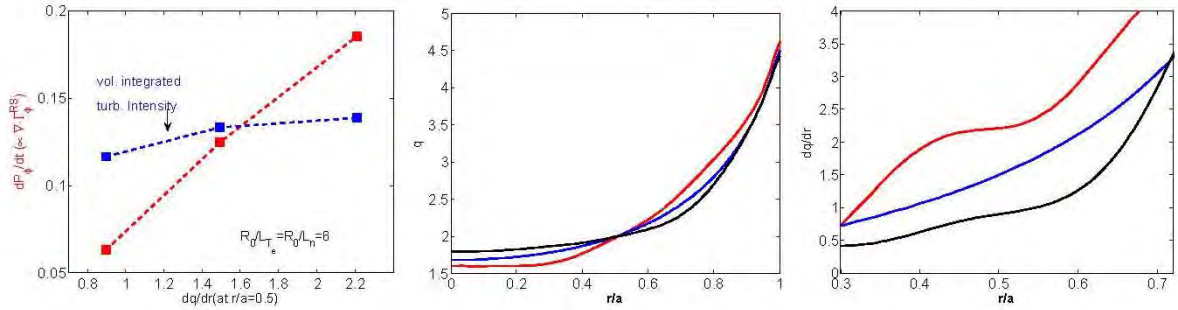


Figure 7. CTEM-driven total intrinsic torque versus magnetic shear averaged over the central core region (left), and corresponding radial profiles of q (middle) and dq/dr (right) of the three equilibria used for these simulations.

magnetic shear. On the other hand, the volume-integrated fluctuation intensity exhibits a much weaker dependence on the magnetic shear, which indicates that the observed intrinsic torque vs the magnetic shear scaling mostly results from the effect of the $k_{||}$ symmetry breaking physics. Specifically, the enhancement of CTEM driven intrinsic torque with magnetic shear is caused by the enhanced $k_{||}$ symmetry breaking due to stronger magnetic shear. The key point of this result is that the dependence of the intrinsic torque on dq/dr indeed produces the right trend which is consistent with the current scaling obtained in Fig. 5. Therefore, given the distinct effects of varying the q value and the magnetic shear on intrinsic torque generation, it is concluded that the current scaling results from the effect of magnetic shear on the turbulence spectrum, namely, the magnetic shear induced $k_{||}$ symmetry breaking, which is enhanced with increased magnetic shear. We should point out that the effect of magnetic shear on the nonlinear residual generation and the associated key role of it behind the current scaling revealed by these gyrokinetic simulations should be tested and validated by experiments. To a certain extent, this can be done by revisiting the experimental data base from which the current scaling was revealed.

For current scan studies, another scenario often adopted in experiments is to hold the q profile and the pressure profile fixed, while varying the current. In this case, the vacuum magnetic field has to change correspondingly, according to $B_{\mathrm{vac}} \propto I_p$. Our nonlinear CTEM simulations have been also carried out to explore the current dependence of intrinsic rotation in this scenario. We used the same simulation parameters as in Fig. 5, except for the MHD equilibria. Simulation results are presented in Fig. 8. In this case, the CTEM driven intrinsic torque is found to increase close to linearly with the vacuum field. It is important to notice that in this scan scenario, the parameter ρ_* for the three cases

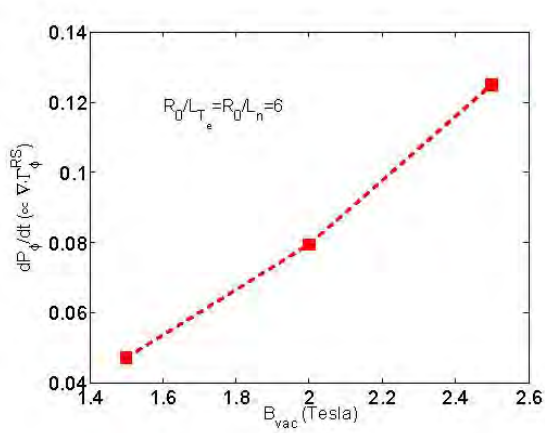


Figure 8. Total intrinsic torque versus external magnetic field B_{vac} in CTEM turbulence.

varies significantly, from $a/\rho_i \sim 130$ for $B_{vac} = 1.5$ Tesla to $a/\rho_i \sim 230$ for $B_{vac} = 2.5$ Tesla, which is an important factor impacting turbulence transport. Thus, the variation in ρ_* should be taken into account when we look into the results in Fig. 8 in connection with the experimental scaling. Nevertheless, this current scan scenario is considered to be less relevant to the current scaling obtained in experiments.

5. Meso-scale Phenomena in CTEM Turbulence—Flow, Particle and Heat Pinch and Large Scale Structure Formation

A few highly remarkable, interesting features observed in our CTEM simulations are discussed in this section.

First, nonlinear GTS simulations have found that meso-scale phenomena and associated nonlocal transport are highly pronounced in the TEM turbulence regime, probably because of strong coherent wave-particle interaction at magnetic precession resonances of trapped electrons. Remarkably, the parallel (and toroidal) flow exhibits coherent temporal burstings and radial propagation during its generation process, as is clearly seen in the upper-left panel of Fig. 9. Particularly, it is shown that small parallel flow perturbations are generated locally (in the center of the plasma in the simulation case) by the turbulence, and then propagate radially. The measured propagation velocity is $\sim 7 \times 10^{-3} c_s$ with c_s the sound speed. This "flow pinch" phenomenon observed in the simulations appears to phenomenologically reproduce a well-known experimental result in JT-60U where perturbed flows created by modulated beams were demonstrated to penetrate radially from the peripheral region of the plasma into the core [3]. Thus, it is highly illuminating. Furthermore, radial pinches appear to be a very robust and generic feature in CTEM turbulence, and are found to emerge in all transport channels, including particle, electron heat and ion heat. These are illustrated in Fig. 9. One highly

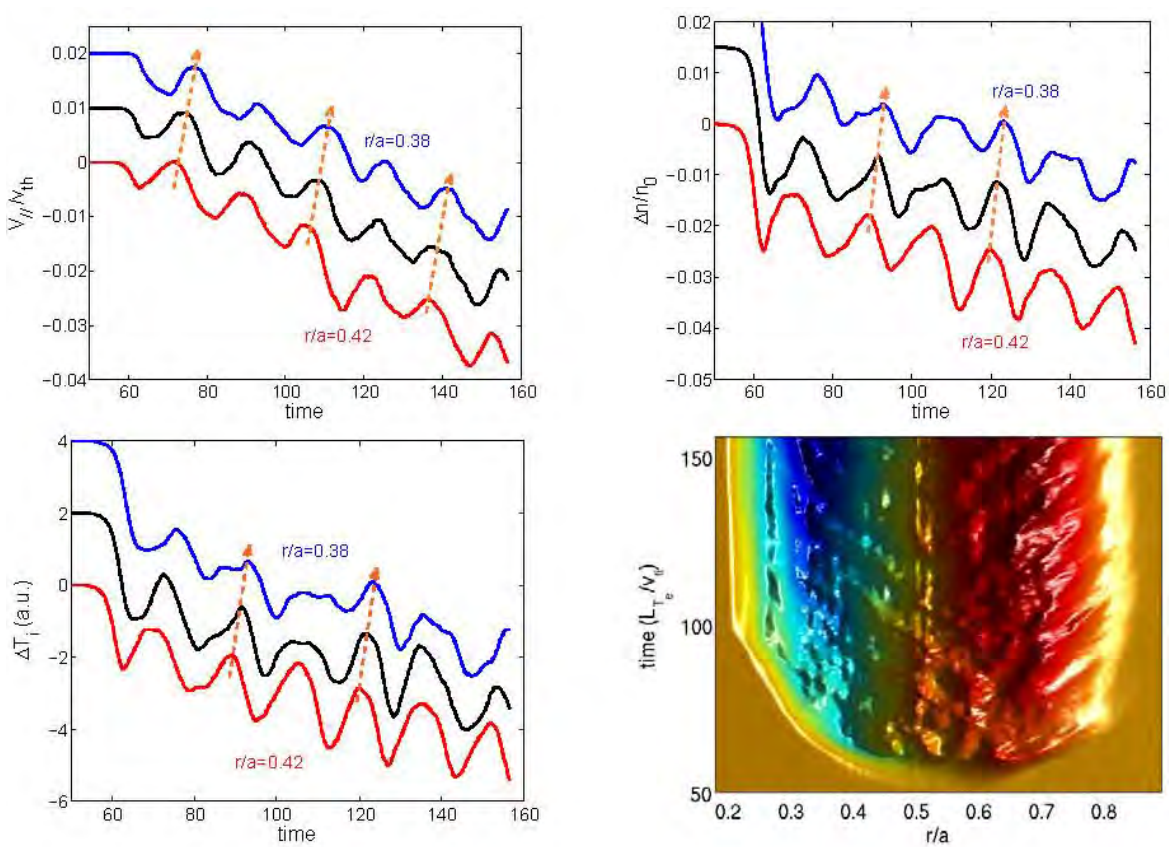


Figure 9. Time history of parallel flow (upper-left), density (upper-right) and ion temperature ΔT_i (lower-left) at three radial locations, and spatio-temporal evolution of electron temperature ΔT_e (lowerright), illustrating generic pinch phenomenon in CTEM turbulence.

remarkable fact found is that the radial pinches in different transport channels emerge "in phase". We point out that the density pinch and heat pinch carried by electron turbulence as suggested by our nonlinear CTEM simulations can be tested by designing similar perturbative experiments to ones with modulated flows.

Finally, as a remarkable meso-scale phenomena, ∇T_e-driven CTEM turbulence in specific, experimentally relevant parameter regimes is found to generate large fluctuation structures with toroidal mode number $n \lesssim 10$ via dramatic inverse energy cascades, as shown in Fig. 10. This phenomenon may have a natural connection to the generation of blobs widely observed in the tokamak edge region.

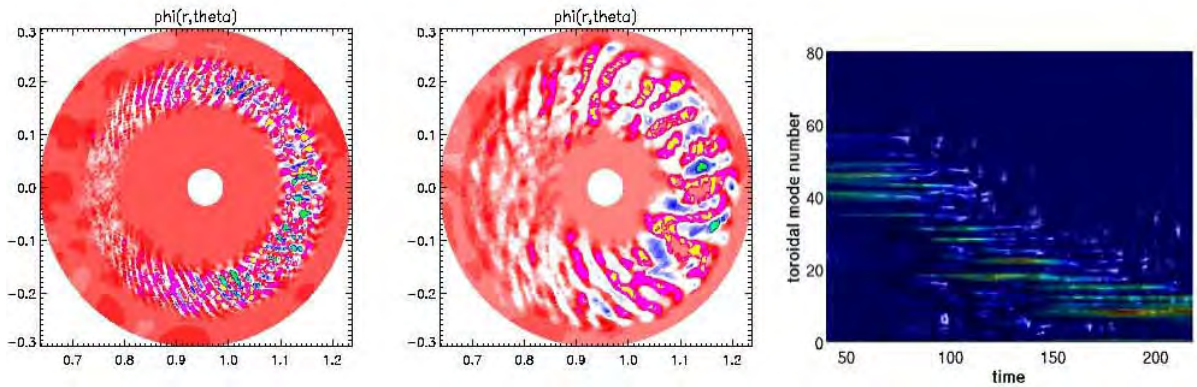


Figure 10. Contour plots of density fluctuations on a poloidal plane at t = 60 (left) and t = 160 (middle), and time evolution of toroidal spectrum $|\delta n_n|^2$ (right) in ∇T_e-driven CTEM turbulence, showing the generation of low-n, blob-like, large fluctuation structures.

6. Summary

State-of-the-art, first-principles-based global gyrokinetic simulations, which exploit leadership-class computational resources and cooperate with experimental and theoretical studies, are shown to lead to real advances in understanding some key physics issues in fusion science.

Specifically, global gyrokinetic simulations using the GTS code have discovered an important nonlinear flow generation process due to residual stress produced by micro-turbulence. The key results remarkably contribute to elucidating critical issues with regard to the origin of intrinsic rotation, underlying physics governing the experimental empirical laws of intrinsic rotation with respect to plasma gradients and current, and underlying dynamics governing the radial penetration of modulated flows in experiments. Simulations also have made important predictions in regimes where experiments can validate.

Acknowledgments

The authors would like to express thanks to Dr. L. Zakharov for generating numerical MHD equilibria using the ESC code for current scan simulations. Simulations were performed on Franklin at the National Energy Research Scientific Computing Center (NERSC) and on Jaguar/pf at the National Center for Computational Sciences (NCCS). This work was supported by U.S. DOE Contract No. DE-AC02-09CH11466 and the SciDAC project for Gyrokinetic Particle Simulation of Turbulent Transport in Burning Plasmas.

References

[1] J. E. Rice *et al*., Nucl. Fusion **44**, 379 (2004).
[2] J. E. Rice *et al*., Nucl. Fusion **47**, 1618 (2007).
[3] M. Yoshida *et al*., Phys. Rev. Lett. **100**, 105002 (2008).
[4] K. Ida *et al*., Nucl. Fusion **50**, 064007 (2010).

[5] S. D. Scott *et al*., Phys. Rev. Lett. **64**, 531 (1990).
[6] P. H. Diamond *et al*., Phys. Plasmas **15**, 012303 (2008).
[7] W. X. Wang *et al*., Phys. Plasmas **13**, 092505 (2006).
[8] W. X. Wang *et al*., Phys. Plasmas **13**, 092501 (2006).
[9] F. L. Hinton *et al*., Phys. Fluids **28**, 3082 (1985).
[10] I. Manuilskiy and W. W. Lee, Phys. Plasmas **7**, 1381 (2000).
[11] W. X. Wang *et al*., Phys. Plasmas **17**, 072511 (2010).
[12] W. X. Wang *et al*., Phys. Plasmas **14**, 072306 (2007).
[13] W. X. Wang *et al*., Phys. Rev. Lett. **102**, 035005 (2009).
[14] R. Dominguez and G. M. Staebler, Phys. Fluids, B **5**, 3876 (1993).
[15] O. D. Gurcan *et al*., Phys. Plasmas **14**, 042306 (2007).
[16] P. H. Diamond *et al*., Nucl. Fusion **49**, 045002 (2009).

Application of 3-D spherical shell adaptive mesh refinement to an atmospheric model with a vertical Lagrangian coordinate

N.G. Andronova, D.R. VandenBerg, Q.F. Stout, R. Oehmke, J.E. Penner, and V. Zubov

Abstract. Here we present an application of a 3D adaptive mesh refinement (AMR) tool to a stratospheric tracer advection in a spherical coordinate system enriched by a new v-scan routine which allows conservation of the vertical integral of meteorological variables. Until now 3-D AMR tools did not take into account the separate tracking of vertical integrals. The latter is crucial to the atmospheric studies where refinement and coarsening routines in the vertical direction change the grid resolution of atmospheric pressure and geopotential height.

1. Introduction

Previously, mesh adaptation was applied to different representations of the Navier–Stokes equations. First, Skamarock et al. (1989) and Skamarock and Klemp (1993) applied their adaptive grid techniques for 3D limited-area models in Cartesian coordinates. Later, Behrens (1998) applied AMR techniques with an adaptive grid to a shallow water model in Cartesian coordinates while Hess (1999) used a grid based on a block-structured data layout. Bacon et al. (2000) and Boybeyi et al. (2001) formulated an adaptive non-hydrostatic regional model based on unstructured, triangulated grids with rotated Cartesian coordinates. Later, this model was used within a formulation with a spherical geometry (Gopalakrishnan et al. 2002, Bacon et al. 2003). There are a few shallow water models with statically adaptive grids developed using a spherical geometry (Ruge et al. 1995, Fournier et al. 2004, Barros and Garcia 2004). The dynamic AMR methods for shallow water models in spherical geometry were introduced by Jablonowski (2004) and Lauter et al. (2007). Most mesh adaptation methods were developed for models with numerical representation based on the finite-volume class of numerical schemes, and thus they all had desirable properties such as conservation and monotonicity.

In this paper we apply AMR methods to the shallow water equations in a spherical geometry with a Lagrangian vertical coordinate (Jablonowski et al. 2006, Penner et al. 2007). It uses the MPI-based communication library (Adaptive Block library for Locally Cartesian Topologies, ABLCarT) to control grid adaptation, particularly in the horizontal x-y direction (Oehmke 2004). The main requirement for the block structure in the library is that every block consists of the same number of grid cells no matter what their actual spatial size, which is consistent with the needs of the finite-volume approach. In this paper we use the 3D version of ABLCartT, enriched by a new functionality that allows conservation of the vertical integral of meteorological variables.

2. Vertical Mesh Adaptation.

Previously ABLCarT, in its 2D formulation, was used in a dynamical core that adapted its grid resolution (either dynamically or statically) in the horizontal direction (Jablonowski et al. 2006, Oehmke et al. 2008). From the library's viewpoint the model's computational grid is viewed as a collection of individual blocks that are independent data units. Each block is a rectangular unit with additional ghost cell regions around its boundaries. Thus, the information from adjacent blocks is

shared to allow computations requiring information from a block's neighbors. A choice of four ghost cells in each direction satisfies the requirements of the parabolic piecewise method (PPM) solution technique, widely used in atmospheric science. All ghost cells are at the same resolution as the parent block. The flow solver is individually applied to all blocks within the grid before ghost cells are updated.

Previously the library was solely applied in the horizontal direction so that the entire vertical column was contained in a single block. However, the spherical grids used to represent phenomena on Earth often need to treat the vertical dimension differently than those in the horizontal direction. For example, atmospheric pressure at a point is determined by the integral of the atmospheric mass above it, and thus is based on calculations that need to be sequenced in a particular order along a specific integration path. Further, each value along the integral is needed to determine the pressure at each cell, not just those at the bottom. This requires that variable information be sent top-down from each adaptive block to the blocks directly below (or above, in other cases), and this must be done (logically) sequentially above each point on the sphere, though it can be done in parallel across all surface points.

To perform vertical integration we have developed a *vertical scan* operation (v-scan) in ABLCarT, which allows computing along vertical paths in the correct order across blocks transparent to their resolution and processor location. To use v-scan the user specifies how the integration (summation) is to be performed in each block, and what information is passed between the blocks, while the library ensures that the operations are performed in the correct order, no matter where the blocks are located. In computer science a *scan* operation is normally used to describe an operation on a single vector (vertical cells above a point on the sphere) rather than on many such vectors in parallel. The ABLCarT performs the v-scan in strict sequential order, so there is no assumption that the operation be commutative or associative.

The 3D refinement involves a large increase in the complexity of the code, but its modular design allows this complexity without extensively altering the user's interface. Data access functions hide lower level implementation details, and adaptation modules are constructed so that each type of change to a block is handled separately and transparently to processor location, allowing the developer to concentrate on one neighbor interaction at a time. The user interfaces are encapsulated so that the library functions such as sphere creation, communication, adaptation, etc., work similarly on 3D and 2D spherical shells.

3. The Stratospheric Tracer Advection

The Brewer-Dobson circulation (BDC) in the stratosphere is a prominent feature of the atmospheric dynamics, which within the stratosphere (10–50 km) carries trace gases from equator to pole. We use an advection test (Zubov et al. 1999) which imitates the BDC with prescribed horizontal and vertical winds. The simplicity and transparency of the test enable us to check whether the model's interpolation subroutines that are "wired" into the Lagrangian vertical transport are not distorted by the v-scan, and that communication of all of the needed variables between the blocks in both the horizontal and vertical directions during grid adaptation is correct.

The prescribed circulation has a closed cell in each hemisphere, which is anti-symmetrical about the equator. The winds are set to zero at the poles and on the upper and lower boundaries, thus the model domain is closed to transport and the total mass of a conservative species should not change in time. The initial species distribution was prescribed to be constant within the rectangle between 32.5°S(N) to 12.5°S(N) in latitude and 19.75 km to 25.75 km in height, with a mixing ratio of 1.6 ppmv, and zero everywhere else. Initially, the 10–50 km vertical domain of the model was subdivided into 0.5 km-thick layers, and the 90S-to-90N latitudinal domain was subdivided into 5° steps. The time step, Δt was set to be 10,041 s (2.8 hr), which satisfies the Courant-Friedrich-Levy (CFL) criterion. The tracer makes one revolution around the center of the circulation cell in about 3 to 4 months.

We performed a few "adaptive runs" using the ABLCarT library. First, we have started with the initial resolution of 40 (levels) and 72 latitudes and used 2 levels of dynamic refinement (40 × 72 × 2)

by applying a prescribed threshold value of 10^{-15} ppmv on the concentration of the tracer in a grid cell. Figure 1 presents the adaptation of the tracer advection at various time steps. Then, we compared the adaptive run with a static high resolution run of 160 vertical levels and 288 points in the latitudinal direction (160 × 288 × 0). Also, we compared our results to an adaptive run with an initial resolution of 80 vertical levels and 144 latitudinal points with one level of refinement (80 × 144 × 0). We found that with the respective initial tracer distribution for the adaptive and high resolution runs, the integrals of the tracer in the horizontal and vertical directions in all three runs correspond to each other, showing that the adaptive transport has the same accuracy as the high resolution run.

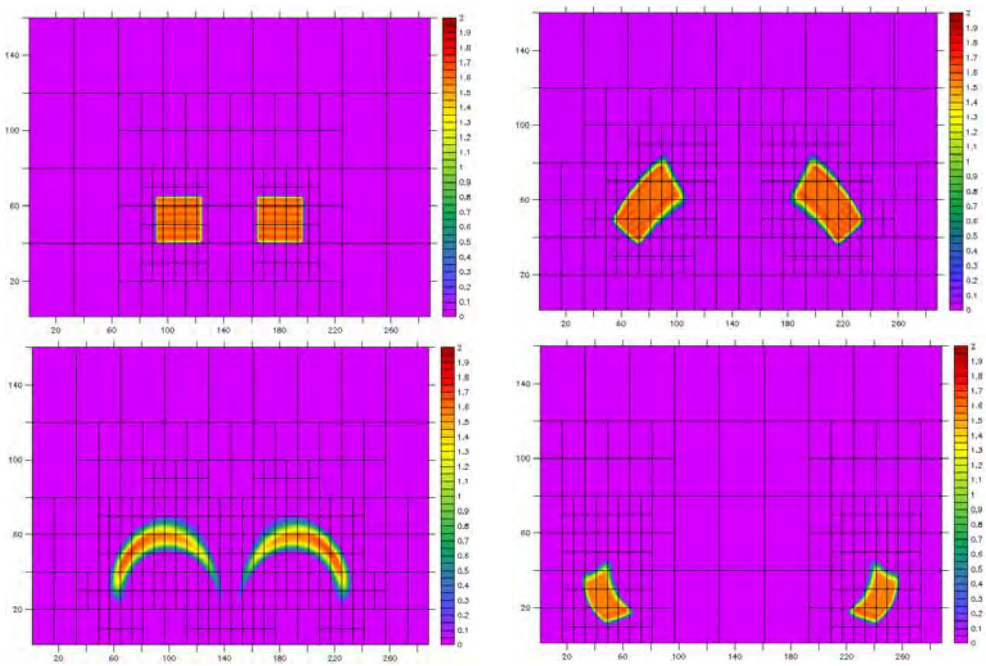


Figure 1. The tracer distribution at different times. Clockwise: days 1, 10, 25, and 100. The wire shows the refinement levels.

One of the important and very favorable aspects of the adaptive model simulation is that accuracy comparable to the static high resolution run can be obtained in less than half the computer time. Furthermore, depending on the specific problem, dynamic refinement could reduce the overall computer time by even more, e.g., if the problem required even higher levels of refinement. Similar reductions are realized in the memory required.

4. Conclusions

We have implemented and tested the three-dimensional parallel adaptive block mesh library (ABLCarT) within the dynamical core of a spherical model with a vertical Lagrangian coordinate. The introduction of a new vertical adaptation feature (v-scan) to the library was a challenging task due to the sequential nature of the vertical atmospheric calculations along a specific integration path. This introduces new geometric configurations of adjacent blocks with differing resolution and the concomitant numerical conservation equations. Applying the ABLCarT with v-scan to a pure advective tracer transport test, we demonstrated that the dynamical mesh adaptation works properly, and gives results equivalent to the refined non-adaptive case.

Acknowledgements

This work was supported through the DOE Scientific Discovery through Advanced Computing program Grant number DOE FG02 01 ER63248.

References

Bacon, D.P., N. Ahmad, Z. Boybeyi, T. Dunn, S.G. Gopalakrishnan, M.S. Hall, Y. Jin, P.C.S. Lee, D.E. Mays, A. Sarma, M.D. Turner, T.R. Wait, K.T. Waight III, S.H. Young, and J.W. Zack, 2003: Dynamically Adapting Unstructured Triangular Grids: A New Paradigm for Geophysical Fluid Dynamics Modeling, *Proc. of Indian Academy of Science*, 69, 457–471.

Bacon, D.P., N.N. Ahmad, Z. Boybeyi, T.J. Dunn, M.S. Hall, P.C.S. Lee, R.A. Sarma, M.D. Turner, K.T. Waight, S.H. Young, and J.W. Zack, 2000: A Dynamically Adapting Weather and Dispersion Model: The Operational Multiscale Environment model with Grid Adaptivity (OMEGA). *Mon. Wea. Rev*., 128, 2044–2076.

Barros, S.R.M., and C.I. Garcia, 2004: A Global Semi-Implicit Semi-Lagrangian Shallow-Water Model on Locally Refined Grids. *Mon. Wea. Rev.*, **132**, 53–65.

Behrens, J., 1998: Atmospheric and ocean modelling with an adaptive finite element solver for the shallow-water equations, *Appl. Numer. Math*., 26, 217–226.

Boybeyi, Z., N.N. Ahmad, D.P. Bacon, T.J. Dunn, M.S. Hall, P.C.S. Lee, R.A. Sarma, and T.R. Wait, 2001: Evaluation of the Operational Multiscale Environment Model with Grid Adaptivity against the European Tracer Experiment. *J. Appl. Met.* 40, 1541–1558.

Fournier, A., M.A. Taylor, and J.J. Tribbia, The spectral element atmospheric model: High-resolution parallel computation and response to regional forcing, *Mon. Wea. Rev.*, **132**, 726–748, 2004.

Jablonowski, C., Adaptive grids in weather and climate modeling, Ph.D. dissertation, University of Michigan, Ann Arbor, MI, 2004, Department of Atmospheric, Oceanic and Space Sciences, 266 pp.

Jablonowski C, Herzog M, Penner J.E, Oehmke R.C, Stout Q, van Leer B, and Powell K.G, 2006: Block-structured grids on the sphere: advection experiments. *Mon. Wea. Rev.*, **134**, 3691–3713.

Gopalakrishnan, S.G., D.P. Bacon, N.N. Ahmad, Z. Boybeyi, T.J. Dunn, M.S. Hall, Y. Jin, P.C.S. Lee, D.E. Mays, R.V. Madala, R.A. Sarma, M.D. Turner, and T.R. Wait, 2002: An Operational Multiscale Hurricane Forecasting System. *Mon. Wea. Rev*., **130**, 1830–1847.

Hess, R., Dynamically adaptive multigrid on parallel computers for a semi-implicit discretization of the shallow-water equations, GMD Tech. Report 9, GMD—German National Research Center for Information Technology, St. Augustin, Germany, 1999.

Nikiforakis, N., 2005: AMR for global atmospheric modelling. In Adaptive mesh refinement—theory and applications (eds T. Plewa, T. Linde, and V. W. Weirs). *Lecture Notes in Computational Science and Engineering*, vol. 41. Berlin, Germany: Springer).

Oehmke, R.C., High performance dynamic array structures, Ph.D. dissertation, University of Michigan, Ann Arbor, MI, 2004, Department of Electrical Engineering and Computer Science, 93 pp.

Penner, Joyce E, Natalia Andronova, Robert C Oehmke, Jonathan Brown, Quentin F Stout, Christiane Jablonowski, Bram van Leer, Kenneth G Powell, and Michael Herzog, 2007: Three dimensional adaptive mesh refinement on a spherical shell for atmospheric models with Lagrangian coordinates *J. Phys.: Conf. Ser*. 78 012072, doi:10.1088/1742-6596/78/1/012072.

Ruge, J.W., S.F. McCormick, and S.Y.K. Yee, 1995: Multilevel adaptive methods for semi-implicit solution of shallow-water equations on a sphere. *Mon. Wea. Rev.*, **123**, 2197–2205.

Skamarock, W.C., and J.C. Klemp, 1993: Adaptive Grid Refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow. *Mon. Wea. Rev.*, **121**, 788–804.

Skamarock, W.C., 1989: Truncation error estimates for refinement criteria in nested and adaptive models. *Mon. Wea. Rev.*, **117**, 872–886.

Zubov, V.A., E.V. Rozanov, and M.E. Schlesinger, 1999: Hybrid Scheme for Three-Dimensional Advective Transport, *Mon. Wea. Rev.*, **127**, 1335-1346.

Subcycled dynamics in the spectral Community Atmosphere Model version 4

Mark A. Taylor,[1] Katherine J. Evans,[2] James J. Hack,[2] and Pat Worley[2]

[1]Sandia National Laboratories, Albuquerque, NM 87185, USA
[2]Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

Email: mataylo@sandia.gov

Abstract. To gain computational efficiency, a split explicit time integration scheme has been implemented in the CAM spectral Eulerian dynamical core. In this scheme, already present in other dynamical core options within the Community Atmosphere Model, version 4 (CAM), the fluid dynamics portion of the model is subcycled to allow a longer time step for the parameterization schemes. The physics parameterization of CAM is not subject to the stability restrictions of the fluid dynamics, and thus finer spatial resolutions of the model do not require the physics time step to be reduced. A brief outline of the subcycling algorithm implementation and resulting model efficiency improvement is presented. A discussion regarding the effect of the climate statistics derived from short model runs is provided.

1. Introduction

This note describes an implementation of a temporal subcycling algorithm in the Community Atmosphere Model (CAM) version 4 Eulerian dynamical core. The CAM is the atmospheric component of the Community Climate System Model (CCSM), a global Earth system model encompassing model components of the atmosphere, ocean, sea ice and land surface. The components communicate through a flux coupler that allows interactions at various time scales set to a level appropriate to capture the relevant model features with accuracy and stability. This core uses a spherical harmonic based spectral method. It requires more computational expense per grid point than other dynamical core options included in CAM, but it remains a favorable option for higher resolution studies due to its isotropic treatment of spherical geometry and the fact that it is well tested with a range of resolutions [1]. It solves the fluid dynamics equations of the model with a semi-implicit time integration scheme, which is a combination of an explicit Robert-filtered leapfrog scheme for the nonlinear terms and a Crank-Nicholson treatment of the linear terms responsible for gravity waves.

2. Subcycling Approach

We start with a high level representation of the equations,

$$\frac{\partial X}{\partial t} + D(X) = F(X) \tag{1}$$

Where $X(t)$ is the state vector at time t, the dynamics terms are represented by $D(X)$ and the forcing tendency terms by $F(X)$. The Eulerian dynamical core uses a process-split approach [2] to couple the dynamics and the forcings tendencies computed by the CAM model physics subroutines. When combined with the semi-implicit time-stepping scheme, (1) is discretized as

$$X(t+\Delta t) = X(t-\Delta t) - 2\Delta t D(X(t-\Delta t), X(t), X(t+\Delta t)) + 2\Delta t F(X(t-\Delta t), 2\Delta t). \quad (2)$$

Most of the terms in the dynamics, D, are evaluated at time t, but because of the semi-implicit scheme there are a few terms that are evaluated at $t - \Delta t$ and $t + \Delta t$. The forcing term, $F(Y, \Delta t_{phys})$ arises from the model physics computed by CAM. We have added a second argument to F because CAM physics can be considered as a function of two variables: a state vector Y and the physics timestep Δt_{phys}. With the semi-implicit scheme used above, the solution is advanced from time $t - \Delta t$ to $t + \Delta t$, representing a duration of $2\Delta t$, and thus with leapfrog time-stepping, the physics timestep is $\Delta t_{phys} = 2\Delta t$. The physics is computed from the state vector $X(t - \Delta t)$ instead of $X(t)$ because the physics contains many dissipative processes that would be unstable if evaluated with the leapfrog scheme. Evaluating that term at time $t - \Delta t$ results in a stable forward-Euler treatment.

For subcycling, we simply reuse the physics tendencies over multiple steps. Let n be the number of subcycled steps and introduce two timesteps, $\Delta t_{phys} = n\Delta t_{dyn}$. A single timestep, representing one physics evaluation and n dynamics steps is then given by
For i=1 ... n:

$$X(t+(i+1)\Delta t_{dyn}) = X(t-(i-1)\Delta t_{dyn}) - 2\Delta t_{dyn} D(X(t-(i-1)\Delta t_{dyn}), X(t+i\Delta t_{dyn}), X(t+(i+1)\Delta t_{dyn})) + 2\Delta t_{dyn} F(X(t-\Delta t_{dyn}), \Delta t_{phys}). \quad (3)$$

That Δt_{phys} is the correct physics timestep to be used as the argument to F can be seen by considering the simplified case with no dynamics D and then writing the final result after n steps in terms of the initial conditions. For n even, we have

$$X(t+(n-1)\Delta t_{dyn}) = X(t-\Delta t_{dyn}) + n\Delta t_{dyn} F(X(t-\Delta t_{dyn}), \Delta t_{phys}) \quad (4)$$

$$X(t+n\Delta t_{dyn}) = X(t) + n\Delta t_{dyn} F(X(t-\Delta t_{dyn}), \Delta t_{phys}) \quad (5)$$

For both final time values $t + (n - 1)\Delta t_{dyn}$ and $t + n\Delta t_{dyn}$, the forcing is applied over a time interval of length $n\Delta t_{dyn}$, and thus it is appropriate to compute the physics with $\Delta t_{phys} = n\Delta t_{dyn}$. For n odd, the situation is slightly different, resulting in

$$X(t+(n-1)\Delta t_{dyn}) = X(t) + (n-1)\Delta t_{dyn} F(X(t-\Delta t_{dyn}), \Delta t_{phys}) \quad (6)$$

$$X(t+n\Delta t_{dyn}) = X(t-\Delta t_{dyn}) + (n+1)\Delta t_{dyn} F(X(t-\Delta t_{dyn}), \Delta t_{phys}) \quad (7)$$

and now the forcing is applied over a time interval of length $(n - 1)\Delta t_{dyn}$ in (6) and an interval of length $(n + 1)\Delta t_{dyn}$ in (7). For this case, we take the average of these two time intervals and set $\Delta t_{phys} = n\Delta t_{dyn}$, matching what was used in the n-even case.

For moisture advection there is one complication due to the fact that the CAM physics returns an adjusted field instead of tendency terms. Let q represent one of the advected moisture fields, such as specific humidity. An implied tendency F_q can be computed from the adjusted field by defining

$$q^{-}(t) = q(t) + \Delta t_{phys} F_q(X(t), \Delta t_{phys})$$

where $q^{-}(t)$ is the adjusted field computed by the CAM physics. With this definition, the original advection scheme

$$\frac{\partial q}{\partial t} + U \cdot \nabla q = F_q(X)$$

is discretized by

$$q^{-}(t-\Delta t) = q(t-\Delta t) + 2\Delta t F_q(X(t-\Delta t), 2\Delta t)$$

$$q(t+\Delta t) = q^{-}(t-\Delta t) - 2\Delta t U \cdot \nabla q$$

which can be written as

$$q(t+\Delta t) = q(t-\Delta t) - 2\Delta t U \cdot \nabla q + 2\Delta t F_q(X(t-\Delta t), 2\Delta t). \tag{8}$$

This equation is now in the same form as (2), and thus to introduce subcycling we follow the identical procedure as was used above.

3. Efficiency Gains with Subcycling

The motivation to incorporate subcycling is to increase efficiency for a forward-in-time simulation with multiple, weakly interacting time scales. Unlike the stability constrained fluid equations in the model, the subgrid scale physics parameterizations can utilize a larger time step size appropriate for capturing the time evolution of a wide range of parameterized physical processes, which for the global atmospheric model in the hydrostatic regime is generally no greater than 30 minutes. The efficiency gain for a simulation using subcycling depends upon the ratio of the size of the dynamics and physics time steps, n, and the percent of time spent evaluating the physics versus dynamics. Simulations of the CAM spectral model run with active atmosphere and land surface components with subcycling have been performed for two resolutions, T85 and T341. T85 and T341 have spatial resolutions about the same as the CAM finite volume dynamical core at one and one-quarter degree resolutions respectively [3]. When T85 and T341 are run with the most favorable configuration for efficiency, which for each is $n = 3$ and 12, the simulations exhibit a decrease in runtime of about 30% and 20% respectively. For the finer T341 resolution, the dynamics portion of the model becomes computationally more expensive although it is being subcycled to a greater degree.

When CAM is used within the fully prognostic Community Climate System Model (CCSM), the efficiency benefit is not simply in the gain of the atmosphere and land model but in the increased efficiency gained by coupling the subcycled model with the rest of the Earth system components. Like the land surface, the sea ice component of the CCSM, CICE [4], is iterated forward in time at the physics time step of the atmospheric model, so CCSM efficiency is gained by the increase in the time step size of the sea ice component. When a fully coupled CCSM version 4 configured for high resolution (T341 atmosphere, quarter-degree finite volume land surface, tenth-degree ocean and sea ice) is run for one full month, the simulation time is reduced from 7.44 hours to 2.97 hours on 6924 processors. Additional gains could be attained by reorganizing processors allocated to each CCSM component now that CICE has become relatively less expensive than the ocean.

4. Climate simulation variation with subcycling

The climate produced by CAM is especially sensitive to the physics timestep [5]. Below we demonstrate this sensitivity using the subcycling capability in T85 CAM spectral Eulerian simulations. The model was tuned without subcycling and using minimum relative humidity thresholds for the formation of low and high stable clouds of 0.915 and 0.68, respectively. The model was run from specified initial conditions for the land surface model rather than a spun up state, which is available with the finite volume land model resolution. The rest of the cloud physics parameters used the default settings for CAM version 4 for all the simulations presented here. Table 1 displays several parameters of interest for two years of simulation using a range of physics time step sizes, Δt_{phys}, of 600 s, 1200 s, and 1800 s. $\Delta t_{phys} = 1200$ s matches the physics time step size of the unsubcycled simulation and $\Delta t_{phys} = 1800$ s is

Table 1. The physics time step size Δt_{phys} is varied for a series of T85 simulations to isolate the its effect on the simulation statistics. 'Nosub' refers to the simulation performed without any subcycling. The number of subcycled dynamics steps, *n*, are 3, 2, and 3, for Δt_{phys} = 600, 1200, and 1800 *s*, respectively. RESTOM, FSNT, CLDLOW, and LWCF refer to the residual energy flux at the model top, the net absorbed shortwave energy flux at the model top, the vertically integrated factional coverage of low clouds, and the longwave cloud forcing respectively (units of W/m^2). 'PBLH' and 'TS' refer to the global annually averaged planetary boundary layer height (m) and surface temperature (degrees K), respectively.

Variable	Nosub (Δt_{phys} = 1200)	Δt_{phys} = 600	Δt_{phys} = 1200	Δt_{phys} = 1800
RESTOM	–0.239	–2.131	0.247	1.613
FSNT	236.634	233.134	236.737	238.705
CLDLOW	34.252	34.755	33.610	32.897
LWCF	28.185	29.719	28.452	27.694
PBLH	609.21	621.93	609.09	599.66
TS	287.406	287.460	287.465	287.390

used in the default finite volume CAM configuration. The global annual net energy exchange at the model top (RESTOM) for 2 years of simulation without subcycling is –0.239, and this was the base value from which we explored variations present in the other model runs. Without subcycling, the physics time step is twice the dynamics time step as explained in Section 2. Since only two years are analyzed, an overall energy balance climate model in this range is reasonable.

For the range of physics time step sizes analyzed, RESTOM and several related variables, such as the net shortwave energy out of the model top (FSNT), the low cloud amount (CLDLOW), and the longwave cloud forcing (LWCF), are linearly dependent on the time step of the physics parameterization scheme. As expected, Δt_{phys} = 1200 *s* is close to the nonsubcycled model. The physics and dynamics time step sizes are the same, so only differences associated with the different algorithm for subcycling outlined in Section 2 and interannual variability are present. The global annual surface temperature (TS) of each simulation is presented to verify that it is invariant to the time step size of the physics and dynamics time step size within expected internal variability of the model (the root mean square of the observational TS error is about 0.7 K). Because TS is not influenced measurably by a variation in the physics time step size and the agreement between the Δt_{phys} = 1200 subcycled and nonsubcycled solutions, we are confident the subcycling algorithm is working as expected.

The origination of this variation of cloud physics parameters with time step size is certainly of interest, and an exhaustive investigation of the details of this dependence would be worthwhile in the improvement of physics parameterization schemes. For our purposes, the source of variations in the fields displayed in Table 1 likely trace back to the planetary boundary layer height (PBLH), which is sensitive to and decreasing with larger physics time step size. Thus, the implied connection between time step size and RESTOM is that the longer time step size results in a lower PBLH, and thus fewer low clouds, which creates a lower albedo and correspondingly a higher absorbed shortwave term and lower value for RESTOM. The less understood feature of these connections is the reason for the lower PBLH with longer time step sizes. One hypothesis is that the longer time step size results in slightly different estimates of boundary layer height (a consequence of the linearizations in the boundary layer parameterization) which can strongly affect the exchange of heat, moisture, and momentum between the surface and free atmosphere. This exercise uncovered unusual behavior in the physics package, specifically wavelike behavior in the instantaneous precipitable water fields, when very short time steps are used.

As with other dynamical cores in CAM, the subcycling introduced into the spectral Eulerian core does require changes to the cloud physics parameters to reproduce the same climate statistics. zonal wind (not shown), vary with physics time step size as well but are expected Thus, the simulations can

produce consistent climate results with the nonsubcycled simulation once the physics parameterization is tuned to account for the longer time step sizes of the physics. In sum, the subcycled physics time step allows more efficient coupled climate simulations without appreciably affecting the climate statistics as defined within the existing CAM physics parameterization scheme.

Acknowledgments
This work was supported in part by the DOE/BER SciDAC project *Modeling the Earth System* and the DOE/BER project *Ultra-High Global Climate Simulation*. This research used the resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. DOE under contract DE-AC05-00OR22725.

References
[1] Collins, W. D., Blackmon, M. L., Bonan, G. B., Hack, J. J., Henderson, T. B., Kiehl, J. T., Large, W. G., McKenna, D. S., Bitz, C. M., Bretherton, C. S., Carton, J. A., Chang, P., Doney, S. C., Santer, B. D., and Smith, R. D. 2006. *J. Climate* **19** 2122–2143.
[2] Williamson, D. L. 2002. *Mon. Wea. Rev.* **130** 2024–2041.
[3] Williamson, D L. 2008. *Tellus* **60A** 839–847.
[4] Hunke, E. C. and Lipscomb, W. 2008. *T-3 Fluid Dynamics Group, Los Alamos National Laboratory, Los Alamos, NM 87545* **LA-CC-06-012**.
[5] Williamson, D. L. 2008. *Tellus* **60A** 848–862.

Earth System Grid Center for Enabling Technologies: Building a Global Infrastructure for Climate Change Research

The Earth System Grid Center for Enabling Technologies Team:

D. N. Williams[3,11], J. Ahrens[4], R. Ananthakrishnan[1], M. Balman[2], G. Bell[3], S. Bharathi[9], D. Brown[6], M. Chen[8], A. L. Chervenak[9], L. Cinquini[5], R. Drach[3], I. T. Foster[1,11], P. Fox[10], S. Hankin[7], D. Harper[6], N. Hook[6], P. Jones[4], D. E. Middleton[6,11], N. Miller[1], E. Nienhouse[6], R. Schweitzer[7], G. Shipman[8], A. Shoshani[2], F. Siebenlist[1], A. Sim[2], W. G. Strand[6], F. Wang[8], C. Ward[9], P. West[10], H. Wilcox[6], N. Wilhelmi[6], S. Zednik[10]

[1] Argonne National Laboratory, Chicago, IL, USA.
[2] Lawrence Berkeley National Laboratory, Berkeley, CA, USA.
[3] Lawrence Livermore National Laboratory, Livermore, CA, USA.
[4] Los Alamos National Laboratory, Los Alamos, NM, USA.
[5] Jet Propulsion Laboratory, National Aeronautics and Space Administration, Pasadena, CA, USA.
[6] National Center for Atmospheric Research, Boulder, CO, USA.
[7] National Oceanic and Atmospheric Administration (PMEL), Seattle, WA, USA.
[8] Oak Ridge National Laboratory, Oak Ridge, TN, USA.
[9] University of Southern California, Information Sciences Institute, Marina del Ray, CA, USA.
[10] Rensselaer Polytechnic Institute, Troy, NY, USA.
[11] E-mail: williams13@llnl.gov, don@ucar.edu, foster@anl.gov

Abstract. Established within DOE's Scientific Discovery through Advanced Computing (SciDAC-) 2 program, with support from ASCR and BER, the Earth System Grid Center for Enabling Technologies (ESG-CET) is a consortium of seven laboratories (Argonne National Laboratory [ANL], Los Alamos National Laboratory [LANL], Lawrence Berkeley National Laboratory [LBNL], Lawrence Livermore National Laboratory [LLNL], National Center for Atmospheric Research [NCAR], Oak Ridge National Laboratory [ORNL], and Pacific Marine Environmental Laboratory [PMEL]), and two institutes (Rensselaer Polytechnic Institute [RPI] and the University of Southern California, Information Sciences Institute [USC/ISI]). The consortium's mission is to provide climate researchers worldwide with a system of federated science gateways to access data, information, models, analysis tools, and computational capabilities required to evaluate peta-scale to exa-scale data sets. Its goals are to (1) make data more useful to climate researchers by developing collaborative technology that enhances data usability; (2) meet the specific needs of national and international climate projects for distributed databases, data access, and data movement; (3) provide a universal and secure Web-based data access portal for broad-based multi-model data collections; and (4) provide a wide range of climate data-analysis tools and diagnostic methods to international climate centers and U.S. government agencies. To this end, ESG-CET is working to integrate in a collaborative problem-solving environment all highly publicized climate data sets—from climate simulations

to observations—using distributed storage management, remote high-performance units, high-bandwidth wide-area networks, and user desktop platforms.

1. The Climate Data Challenge and the Earth System Grid

Climate change research places high demands on data-management, -analysis and -storage resources. These sources reside on distributed national and international data holdings. Large-scale processing is required to compare observations, which vary greatly in their temporal and spatial sampling, with simulated outputs from models. As data has become increasingly abundant and complex, transporting and effectively processing it has also become increasingly difficult. For many scientific disciplines, the amount of data being generated, inspected, and studied ranges up to tens of petabytes, with estimates for the near future breaking the exascale barrier (ASCR Science Network Requirements, 2009). At these data sizes, transferring data is prohibitively expensive; instead, the simpler approach is to move the computation to where the data sets are stored. Based on this approach, the Earth System Grid (ESG) (Williams, D. N., 2009) has developed a system of gateways and data nodes where stored data can be accessed and analyzed in place. Gateways provide climate researchers a doorway (e.g., via a Web portal) to data, information, models, analysis tools, and computational resources, typically providing an integrated view of data in many data nodes; data nodes are located where the data resides and are responsible for providing data access services.

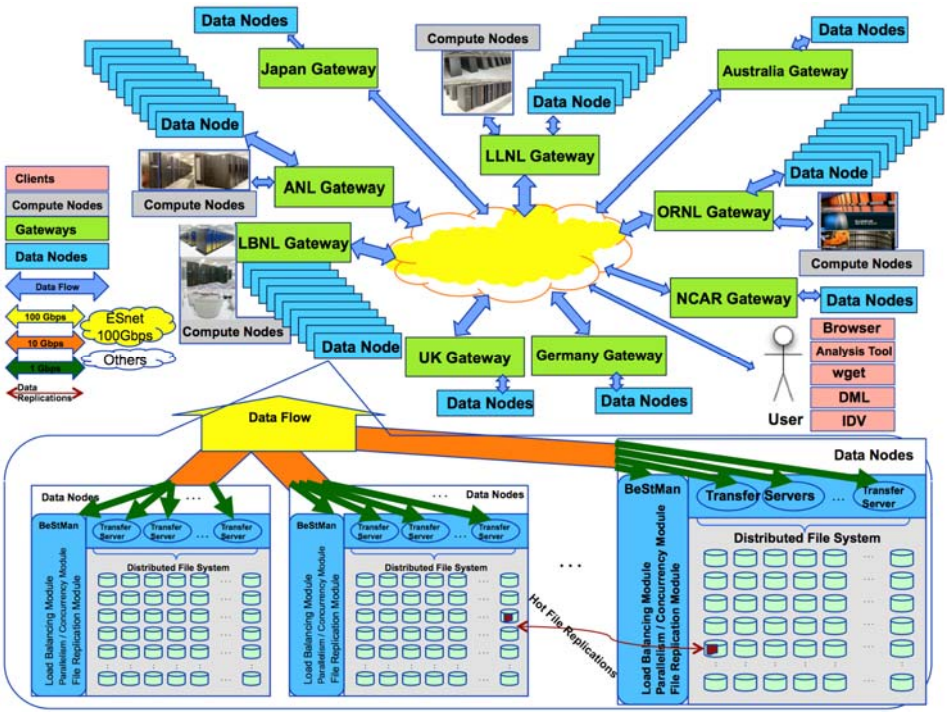


Figure 1: The topology of the ESG enterprise system with network connections and computing resources, which provides a network of geographically distributed gateways, data nodes, and computing resources in a globally federated, built-to-share scientific discovery infrastructure.

Figure 1 shows ESG ongoing partnerships and established relationships with a variety of data, research, and technology efforts. To meet the needs of climate community, these collaborations will be augmented with additional observational and model-generated data sets, including those from the Intergovernmental Panel on Climate Change (IPCC) Coupled Model Intercomparison Project, phase 5 (CMIP5), the Community Climate System Model version 4 (CCSM4), the Community Earth System

Model version 1 (CESM1), the Carbon-Land Model Intercomparison Project (C-LAMP), the Parallel Ocean Program (POP), the North American Regional Climate Change Assessment Program (NARCCAP), the National Aeronautics and Space Administration (NASA) satellite observational data (CloudSat, and others), the National Oceanic and Atmospheric Administration (NOAA) observational data, and many other data sets critical to understanding the state of the climate and changes to it.

In addition to these data, ESG is unifying computational and analytical climate resources under one unique powerful knowledge-discovery enterprise system. ESG continues to evolve to maintain its reputation as a reliable resource for serving the climate-science community as it seeks to derive fundamental and practical understanding of climate and climate change.

2. Volume of Data Downloaded

ESG has become an important resource for a number of climate initiatives (e.g., CMIP5, CCSM, NARCCAP, NASA observations, and more as listed above) that are central to climate investigations. In order to access and download data, ESG users must create an account at an ESG gateway. Many data (e.g., CMIP5, CCSM, POP, NARCCAP, C-LAMP, AIRS, etc.) are available for free download at one of the four ESG current gateways, located at the Program for Climate Model Diagnosis and Intercomparison (PCMDI), http://pcmdi3.llnl.gov/esgcet/home.htm; NCAR, http://www.earthsystemgrid.org/; ORNL, http://esg.ccs.ornl.gov/esgcet/home.htm; and NASA/JPL, http://esg-gateway.jpl.nasa.gov/home.htm. Figures 2, 3, and 4 illustrate the growth in ESG usage over the last five years. Data download volumes routinely total in the tens of terabytes (TB, or 1×10^{12} bytes) per month. The cumulative total since January 2005 reached **one petabyte** (PB, 1×10^{15} bytes) in February 2010. The total number of registered users by the end of 2010 is expected to exceed 20,000. In a given month, between 500 and 800 users are active—up to 30 or more per day.

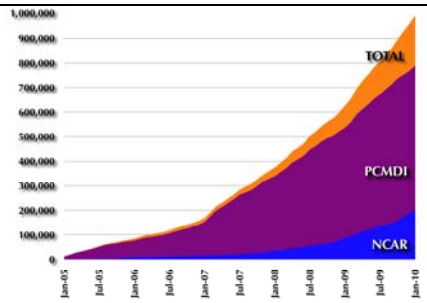


Figure 2: Cumulative ESG download volume, in GB.

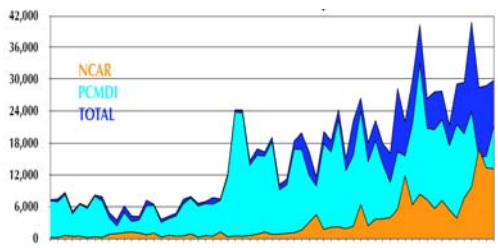


Figure 3: Per-month ESG download volumes, in GB.

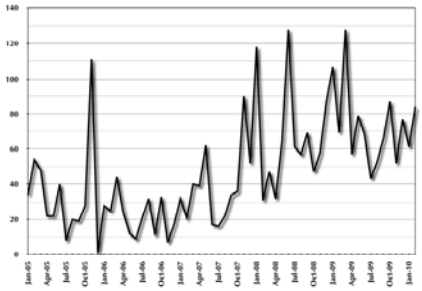


Figure 4: Number of emails per month to the ESG-CET support (esg-support@earthsystemgrid.org). An average day sees between 3 and 4 emails, with spikes correlated to releases of new data sets and codes. Many emails are questions regarding the served data; others are standard support questions involving resetting usernames and passwords. We see that the user community is well engaged with ESG-CET.

3. ESG-CET is Positioning for CMIP5 (IPCC AR5) Data

Just as ESG was critical to the success of CMIP3 (Williams et al., 2009), ESG has also been given the responsibility to meet the data management needs of CMIP5, which will provide results informing the IPCC's Fifth Assessment Report (AR5) (IPCC 2007). Under the leadership of PCMDI at LLNL and with the help of NCAR, ORNL, and others in the national and international community—including centers in the United Kingdom (U.K.), Germany, Japan, and Australia—an internationally federated, distributed data archival and retrieval system is being established. The CMIP5 data archive will dwarf that of its predecessor (CMIP3—"4" was skipped in CMIP numbering in order to align with AR numbering). Instead of the 35 TB collected for CMIP3, modeling groups will generate tens of PB. A subset of results (~2 PB) that is expected to be of highest interest to researchers will be replicated at PCMDI and several other data centers. PCMDI will supervise the distributed CMIP5 data archive and oversee the effort to provide access to this valuable collection of output from model simulations and observations.

CMIP5 model output will mainly be made available through ESG data nodes located at modeling centers, but PCMDI will serve as a hub for access to this distributed data archive. To further enhance accessibility to this subset of the data, it will subsequently be replicated at additional global sites (in the U.K., Germany, Japan, and Australia), where it will be served through gateways coordinated throughout the ESG network (see Figure 1). In responding to this challenging undertaking, PCMDI has organized partnerships with global data centers funded to assist with CMIP5 data retrieval and dissemination and to create an internationally distributed data archival and retrieval federation, known as the Earth System Grid Federation (ESGF). The federation utilizes software primarily developed by the ESG-CET.

Ongoing partnerships and established relationships to a wide variety of data, research, and technology efforts position ESG to continue to speed progress in climate science. Although in many cases ESG tools and technologies primarily benefit the climate community, ESG tools have sometimes proved to be general enough to serve other science communities.

4. Gateway Data Discovery and Access

The development of the software stack that will run on ESG gateways has been guided by two main goals: the imminent transition of portals running ESG software at PCMDI, NCAR, ORNL, and NASA/JPL to the new infrastructure, and support for the upcoming CMIP5 model output streams. Gateway functionality has been augmented and revised in many respects, including the following major areas of development:

- **Faceted search.** The ESG-faceted search module is a search interface that allows easy and rapid browsing of data sets in a way that never allows the user to feel lost in the data. A completely new user interface for the ESG-faceted search has been developed, and featured prominently on the home pages of the PCMDI and NCAR gateways. The new interface follows paradigms that are becoming common across many e-business Web sites, thus facilitating the user experience as well as increasing the scalability and maintainability of the application.

- **Data Reference Syntax (DRS).** The DRS is a common naming system to be used in files, directories, metadata, and URLs to identify data sets wherever they might be located within the distributed CMIP5 ESGF archive. The gateway code can now ingest and expose metadata conforming to the DRS specification that will be used to organize and index the CMIP5 data archive. This includes parsing of the DRS information from the Thematic Realtime Environmental Distributed Data Services (THREDDS) catalogs, persistence in the relational database, and configuration of DRS facets in the metadata query interface.

- **Data versioning.** The underlying distributed data model has been heavily re-factored to support the tracking of multiple versions of data. Thus, data publishers can publish, update, and retract new versions in response to quality control results. Versioning support had implications at all

levels of the gateway software: the relational object model, the publishing services, the metadata query services, and the user interface.

- **Data migration.** A procedure for migrating the relational database content when upgrading the underlying schema has been set up based on the open-source Liquibase project. This procedure addresses a critical operational need since it will allow gateway administrators to upgrade to a new major software release without having to republish all the data or ask users to register again.
- **Attribute and Authorization service.** The gateway application has been augmented with attribute and authorization services, which can deliver Security Assertion Markup Language (SAML)-signed assertions in response to attribute and authorization queries by remote clients respectively, facilitating a fine-grained attribute based authorization policy enforcement. The data node plug-ins for the GridFTP (Allcock et al., 2005) and THREDDS Data Server (TDS) servers leverage these services to determine access to requested data.

  ESG, in collaboration with its European partners, chose SAML as the enabling technology to exchange user attributes and access control information among sites. Each ESG gateway or partner data center will deploy a SAML-based attribute service, which other gateways can securely query to retrieve attribute information about a specific user. This information is required to authorize users both to access specific data sets controlled by a group at another site and to store complete data access metrics.
- **Metadata exchange.** The gateway infrastructure for metadata exchange, based on the Open Archives Initiative's Protocol for Metadata Harvesting (OAI-PMH), has been updated to support versioning and replication, as well as the capability to execute selected harvesting by project so that a gateway needs only import those records that are relevant to its user community, and not others.
- **Data download.** The user workflows for selecting and downloading the data have been thoroughly revised to make them more friendly, efficient, and supportive of a variety of options, including selection across data collections and gateways, retrieval of files from deep storage via BeStMan, generation of wget scripts and integration with the DataMover-Lite (DML) desktop client.
- **Model metadata.** The Earth System Curator project has continued to work with ESG to expand the gateway functionality for ingesting and servicing model metadata, including the capability to connect data sets to the models and simulations that produced them, full handling of the CMIP5 conformance properties, ingestion of Common Information Model (CIM) metadata from the Common Metadata for Climate Modelling Digital Repositories (METAFOR) Questionnaire application, and several improvements to the user interface (the model “trackback” pages). Progress in this area was guided by a series of demonstrations that the Curator project organized to solicit continuous feedback from national and international partners, such as the Global Organization for Earth System Science Portals (GO-ESSP) and METAFOR communities.
- **Federated authentication.** ESG is designed as a federated system allowing user access via the ESG gateways and supporting interoperability with other non-ESG partner data centers (Siebenlist et al., 2009). The ESG infrastructure leverages the OpenID protocol, which supports cross-site authentication between the many gateways as well as with European Web portals. OpenID is a single-sign-on technology that allows users to register and create an identity at only one site, where their credentials are stored, and then carry their authenticated identity as they navigate and access data throughout the ESG federation. All web-based access to data node services leverage this authentication protocol for end user authentication. Some data node services require the use of Public Key Infrastructure (PKI). ESG gateways operate Credential

Translation Services that allow a user to use a single OpenID login to obtain PKI credentials. This approach results in a seamless access experience for the end user.

- **Gateway user interfaces.** The gateway user interface has been revised and improved throughout the site, including major changes in the pages for metadata search, data download, model trackback, user and group administration.

The gateway software has been deployed, tested, and officially release to the community according to a carefully planned schedule aimed at progressively enhancing the functionality of the application, while at the same time soliciting feedback from a progressively larger user base.

5. Data Node Server-Side Analysis

In the ESG data node software stack (Figure 5), we have begun the development of a coordinating entity called the Data Node Manager (DNM). The architecture of this component is distinguished from that of other components in that it must address cross cutting concerns that span all constituent elements of the stack. The DNM logs, monitors, and collects metrics over the entire software stack. It has been designed to be highly fault tolerant and perform under load with graceful degradation properties.

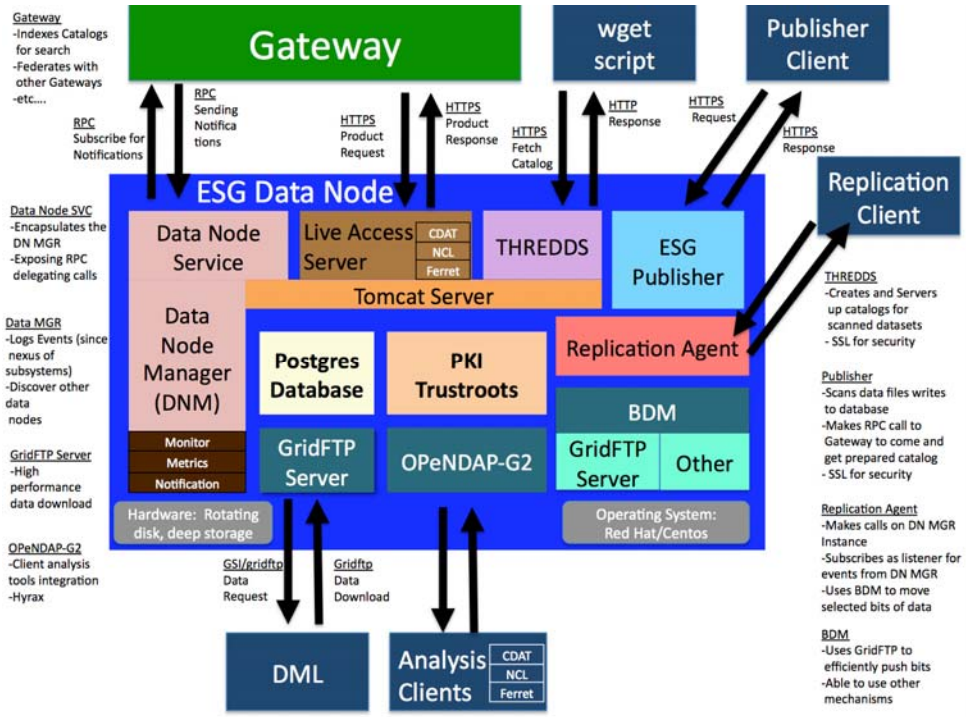


Figure 5: The operations performed by the data node within the ESG infrastructure.

In the ESG infrastructure, the actual data holdings reside on a multiple federated data nodes. In addition to hosting those data, a data node includes the metadata services needed to publish data to gateways and execute data-product requests through these portals. Personnel can set up nodes at local institutions, and a single ESG gateway can serve data requests to many associated nodes. For example, more than 30 institutions are expected to operate data nodes as part of the IPCC CMIP5.

The ESG infrastructure includes the following components, which are also shown in Figure 5:

- **Support for high-performance data download client.** The gateway application is augmented with the DML client, which supports high-performance, multiple-file download. Users can

download and configure DML directly through the gateway and then start it via a file list in XML format. With this system, users can request access to files stored on a local rotating disk or those in deep storage archive, which are then transferred to the gateway cache.

- **Data replication.** Data replication is an important facet of international collaborations such as ESG. No one facility can host the accumulated holdings for the data-intensive climate research community; therefore, archives at one institution are replicated at a collaborator's site via software links. ESG provides a robust data replication client, developed and improved by analyzing replication use cases and requirements, to ensure that the IPCC archive stored at LLNL is available to all participating European, Asian, and Australian data centers.
- **Modeling metadata.** The ESG collaboration is working with the Earth System Curator (ESC) project at developing the full infrastructure so that it accurately captures and displays model metadata within the gateway Web application.
- **Data publishing operations.** After the data node software stack is installed, users can run software scripts required to publish all current ESG data holdings (i.e., climate model simulations and observations) to the gateway. This support for full publication ensures that users have access via the data node software stack application to all holdings, including nonstandard model runs, multiple deep-storage archives, and multiple data access services.

The ESG data node software stack is distributed nationally and internationally via a script or a virtual machine (VM), a software platform that allows a complete operating system (a guest OS) to run as an application inside another operating system (the host OS). The guest OS executes software applications identically to a physical machine. The VMs contain the fully installed CentOS (with requisite libraries and other functionality installed) and the data node software stack. This setup eliminates the need for the system to check all software prerequisites. Hardware virtualization via VM improves security, insulates the hardware from attack and user error, offers code portability and ease of backup, and protects the system against potential software conflicts.

6. Data Node Security Infrastructure

ANL, NCAR, and the British Atmospheric Data Centre (BADC) have collaborated closely to define and implement a new ESG security architecture to support secure data access by browsers and rich analysis and visualization clients throughout the federation. The new architecture is based on technologies such as OpenID, SAML, and X509 certificates and will replace the current ESG token-based access. As part of this collaboration-wide effort, ESG has delivered the following components:

- A Java-based OpenID Relying Party (ORP), which is a Web application (running within a Tomcat servlet container) that is responsible for either validating a user certificate presented via SSL authentication or redirecting a user browser to an OpenID Identity Provider, with the net effect of establishing a secure authentication cookie that can be used by other data node software components as a proof of the user's identity.
- A servlet filters infrastructure that can be used in conjunction with the ORP application to secure a generic Java-based data server, such as the THREDDS Data Server that will be part of the standard ESG data node application stack. Existing implemented components include an authentication filter to consume the cookie set by the ORP and an authorization filter that contacts the gateway authorization service.
- Data download clients that can be used with the new security infrastructure, including wget scripts that use client certificates to securely download the requested file by using SSL for authentication. A Java Webstart application client for the Credential Translation Service, which is a MyProxy Online CA, will also be integrated to allow users to download credentials to their local machine, to use with DML clients.

7. Bulk Data Mover

The Bulk Data Mover (BDM) is responsible for the successful transport of large data sets. Climate data sets are characterized by large volume of files with extreme variance in file sizes. The ESG uses BDM as a higher-level data transfer management component to transfer files in data sets into ESG data nodes. BDM achieves high performance using a variety of techniques, including advance transfer queue management, multi-threaded concurrent transfer connections, data channel caching, load balancing over multiple transfer servers, adaptive transfer adjustment, and dynamic parameter tuning.

BDM can accept a request composed of multiple files or an entire directory. The files or directory are described as Universal Resource Locators (URLs) that indicate the source sites that contain the files. The request also contains the target site and directory where the replicated files will reside after successful transfers. If a directory is provided at the source, then BDM will copy the structure of the source directory at the target site. BDM is capable of transferring multiple files concurrently as well as using parallel transmission control protocol (TCP) streams. The optimal level of concurrency or parallel streams is dependent on the bandwidth capacity of the storage systems at both ends of the transfer as well as achievable bandwidth on the wide-area-network (WAN). Setting up the level of concurrency correctly is an important issue, especially in climate data sets, because of the small files. Concurrency that is too high becomes ineffective (high overheads and increased congestion), and concurrency that is too low will not take advantage of available bandwidth. A similar phenomenon was observed when setting up the level of parallel streams in a single file transfer, such as GridFTP.

BDM is designed to work in a "pull mode", where it runs as a client at the target site. This choice is made because of practical security aspects: site managers usually prefer to be in charge of pulling data rather than having data pushed at them. However, BDM could also be designed to operate in a "push mode" or as an independent third-party service. Because large-scale data replication can take a long time (from many minutes to hours and even days), BDM must be an asynchronous service. That means that when a replication request is launched, a request token is returned to the client. The client can use that request token to check the status of the request execution at any time. Another obvious implication of the long lasting nature of large-scale replication is the need for automatic monitoring and recovery from any transient failures.

8. Multi-Phase Transfer Request Management

The tasks that BDM performs to accomplish successful replication are organized into three phases, as shown in Figure 6. The initialization phase plans and prepares file replications from the data source to the local target storage. It includes the following tasks: 1) storage allocation verification at the target site—this requires collecting the total data size from the source site; 2) generating a request plan—the plan includes the initial level of concurrency, the number of parallel streams, and the buffer size for the request; 3) returning initial request estimation to the client; and 4) mirroring the directory structure of the source at the target site. It then generates an execution plan that includes pair-wise source-to-target URLs for all the files to be replicated. This is used by the execution phase.

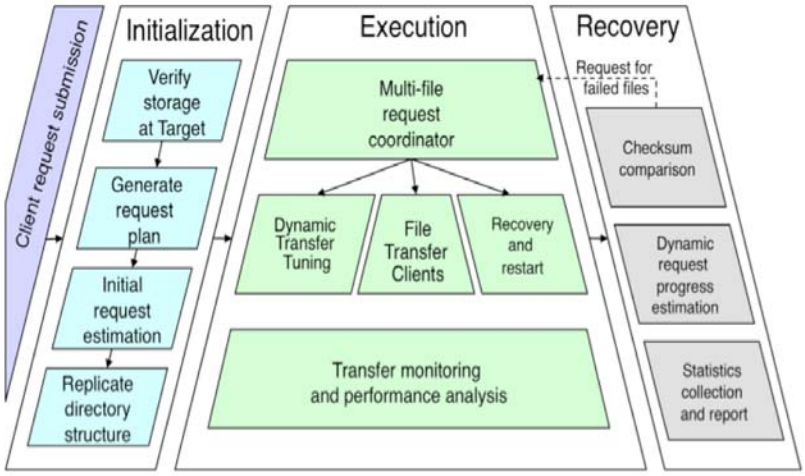


Figure 6: Design of the Bulk Data Mover

The execution phase transfers the requested files, while monitoring and analyzing transfer performance for dynamic adjustment of the transfer properties. It consists of four modules: 1) the multi-file request coordinator uses the information from the "execution plan" and transfer properties including the concurrency level, and accordingly instantiates the file transfer client; 2) the file transfer client performs the transfers – it can support any transfer protocols or services preferred by the virtual organization, including GridFTP, HTTPS, secure copy (SCP), and secure FTP (SFTP); 3) the recovery and restart module continuously monitors the health of the system and the files being transferred, and attempt to re-transfer files that failed to transfer properly; and 4) the module responsible for monitoring and adjusting concurrency collects dynamic transfer performance, and if significant discrepancies from the estimated performance are noticed, adjusts the number of concurrency and parallel streams.

The recovery phase interacts dynamically with the components of the execution phase to validate the completed request by collecting statistics, generating dynamic progress estimation on-demand, and validating transferred files at the end of the request. It has three functions: 1) it collects statistics from the execution of the replication request; 2) it generates dynamic progress estimations on-demand when a client asks for request progress status, which means that this module needs information on file transfers that completed, are in-progress, or are pending, as well as bandwidth usage statistics and estimation; and 3) it validates files, so this module can be running as soon as files are transferred, or at the end of the request, depending on the site preference. The reason for preferring file validation by checksum comparison after all transfers are complete is that calculating checksums is computationally intensive and may disturb the running transfers. This module is also responsible for re-submitting files whose checksums indicated data corruption.

9. Product Services

ESG is intended to serve information products to users of widely varying sophistication—from numerical modelers, who want access to "raw" model output files and verbatim subsets of model output; to climate impacts investigators, who often want rapid access to these small subsets of data without the complexities of model-specific coordinate systems; and to users who only want to quickly visualize the overall behaviours of models. Information products such as scientific visualizations, tables and results from simple statistical calculations are provided by the Live Access Server (LAS) developed at PMEL. Highlights of the enhancements to LAS made through ESG support are:

- **LAS security integration.** Ongoing work focussed on integrating security solutions with LAS will result in the LAS deployments leveraging ESG-CET security infrastructure for authentication and authorization of user access.

- **Data intercomparison capabilities through the vizGal interface**. This interface provides the ability to assess differences both qualitatively (via visual inspection) and quantitatively (via regridding and differencing) between fields. Intercomparison tools may be utilized along slices in any dimension, including comparison of maps, time-series, Hoffmuller plots, and vertical sections (see Figure 7).
- **THREDDS catalog "cleaning tools."** An important part of the ESG-CET science mission is to enable meaningful intercomparisons between disparate data collections. Unfortunately, many interesting data sources to which models can be compared (i.e., satellite data collections, operational model outputs, and gridded observational assimilations) are not kept in well-organized archives with fully developed file-level metadata. To help organize such data collections, we have built THREDDS catalog scanning tools that can identify data sets with excellent metadata from large collections, automatically create aggregations from collections of individual time-series files, and create catalogs that reflect only the best of large, often jumbled data repositories. These clean catalogs can let scientists from ESG-CET and elsewhere quickly identify data sets that can be used to further their scientific goals.

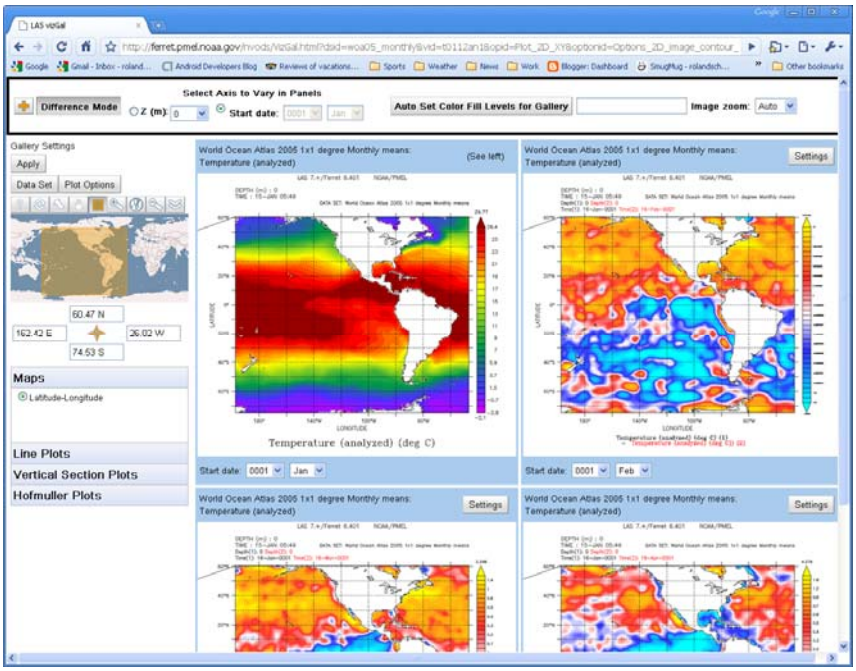


Figure 7: Screen image of the vizGal interface showing differences between temperature fields.

10. Conclusions

The climate research community as a vital infrastructure component relies heavily upon ESG. By unifying computational and analytical climate resources under one unique and powerful knowledge discovery enterprise system, it has enabled research that would otherwise have been performed only with great difficulty or perhaps not at all. ESG continues to evolve to maintain its reputation as a reliable resource for serving the climate science community as it seeks to derive fundamental and practical understanding of climate and climate change.

11. Acknowledgements

This work is supported through the U.S. Department of Energy Office of Science, Offices of Advanced Scientific Computing Research and Biological and Environmental Research, through the SciDAC program. Argonne National Laboratory is managed by Argonne Chicago LLC under Contract DE-AC02-06CH11357. Information Sciences Institute is a research institute of the Viterbi School of

Engineering at the University of Southern California. Lawrence Berkeley National Laboratory is managed by the University of California for the U.S. Department of Energy under contract No. DE-AC02-05CH11231. Lawrence Livermore National Laboratory is managed by the Lawrence Livermore National Security, LLC for the U.S. Department of Energy under contract No. DE-AC52-07NA27344. Los Alamos National Security is managed by LLC (LANS) for the U.S. Department of Energy under the contract No. DE-AC52-06NA25396. National Center for Atmospheric Research is managed by the University Corporation for Atmospheric Research under the sponsorship of the National Science Foundation. Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the U.S. Dept. of Energy under contract DE-AC-05-00OR22725. Pacific Marine Environmental Laboratory is under the National Oceanic and Atmospheric Administration's line office of Ocean and Atmosphere Research, lies within the U.S. Department of Commerce.

The ESG-CET executive committee consists of Dean N. Williams, LLNL; Ian Foster, ANL; and Don Middleton, NCAR.

12. References

[1] ASCR Science Network Requirements, 2009: Office of Advanced Scientific Computing Research Network Requirements Workshop 2009. http://www.er.doe.gov/ascr/ProgramDocuments/Docs/ASCR-Net-Req-Workshop-2009-Final-Report.pdf.

[2] IPCC (Intergovernmental Panel on Climate Change), 2007: Climate Change 2007. www.ipcc.ch.

[3] Williams, D. N., 2009: “The Planet at Their Fingertips: Climate Modeling Data Heats up,” SciDAC Review, 9. http://www.scidacreview.org/0902/html/esg.html.

[4] Williams, D. N., et al, 2009: “The Earth System Grid: Enabling Access to Multi-Model Climate Simulation Data,” Bulletin of the American Meteorological Society, January 2009.

[5] Allcock, B., Bresnahan, J., Kettimuthu, R., Link, M., Dumitrescu, C., Raicu, I. and Foster, I., 2005, "The Globus Striped GridFTP Framework and Server." SC'2005.MyProxy Credential Management Service, http://grid.ncsa.uiuc.edu/myproxy/

[6] OpenID, http://openid.net/

[7] Security Assertion Markup Language (SAML), OASIS Security Services (SAML) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[8] Bulk Data Mover, http://sdm.lbl.gov/bdm/

[9] DataMover-Lite, http://sdm.lbl.gov/dml/

[10] Siebenlist, F., Ananthakrishnan, R., Bernholdt, D.E., Cinquini, L., Foster, I., Middleton, D., Miller, N., and Williams, D.N., 2009. “Enhancing the earth system grid security infrastructure through single sign-on and autoprovisioning,” Workshop on Grid Computing Environments.

MAESTRO, CASTRO, and SEDONA—Petascale codes for astrophysical applications

A. Almgren,[1] J. Bell,[1] D. Kasen,[2] M. Lijewski,[1] A. Nonaka,[1] P. Nugent,[1] C. Rendleman,[1] R. Thomas,[1] M. Zingale[3]

[1]Lawrence Berkeley National Laboratory
[2]UC Santa Cruz
[3]Stony Brook University

Abstract. Performing high-resolution, high-fidelity, three-dimensional simulations of Type Ia supernovae (SNe Ia) requires not only algorithms that accurately represent the correct physics, but also codes that effectively harness the resources of the most powerful supercomputers. We are developing a suite of codes that provide the capability to perform end-to-end simulations of SNe Ia, from the early convective phase leading up to ignition to the explosion phase in which deflagration/detonation waves explode the star to the computation of the light curves resulting from the explosion. In this paper we discuss these codes with an emphasis on the techniques needed to scale them to petascale architectures. We also demonstrate our ability to map data from a low Mach number formulation to a compressible solver.

1. Introduction

We present a suite of codes for studying astrophysical phenomena whose target is the end-to-end simulation of a Type Ia supernova (SN Ia) at the petascale. Each code is designed to perform optimally for a particular flow regime. For the early convective phase of a carbon/oxygen white dwarf leading up to ignition, we use MAESTRO [3], a hydrodynamics code based on a low Mach number approach that allows long-time integration of highly subsonic flow. The time step in MAESTRO is controlled by the fluid velocity instead of the sound speed, allowing a much larger time step than would be taken with a compressible code. Once the star ignites and the fluid begins to travel at speeds no longer small relative to the speed of sound, the low Mach number assumption is invalid and the fully compressible equations must be solved to simulate the final seconds of stellar evolution before the explosion. We simulate the explosion phase of SNe Ia with CASTRO [1], a fully compressible hydrodynamics code. Finally, SEDONA [2], a multidimensional, time-dependent, multi-wavelength radiation transport code, is used to calculate the light curves and spectra from the resulting ejecta, enabling direct comparison between computational results and observation. All three codes have been designed to harness the resources of the most powerful supercomputers available, and scale well to 100k–200k cores.

MAESTRO and CASTRO use structured grids with adaptive mesh refinement (AMR); SEDONA uses an implicit Monte Carlo approach. A time step in CASTRO requires the fully explicit advance of a hyperbolic system of conservation laws, as well as the computation of self-gravity. A time step in MAESTRO is composed of explicit advection as well the solution of a variable-coefficient Poisson equation that follows from the velocity constraint resulting from the low Mach number approximation. A time step in each code also involves evaluations of the equation of state as well as computation of

any reactions. In addition to simulations of SNe Ia (see Figure 1), CASTRO is also being used to study core-collapse and pair-instability supernovae, and MAESTRO is being applied to convection in massive stars, X-ray bursts, and classical novae.

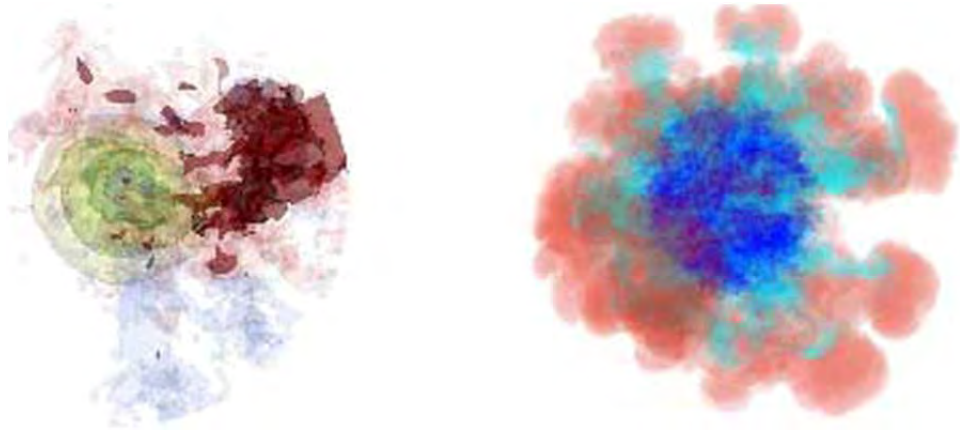


Figure 1. (Left) MAESTRO simulation of convection in a white dwarf preceding a SN Ia. Shown are contours of radial velocity (red = outward, blue = inward) and nuclear energy generation. This simulation was performed using Jaguar at OLCF with an effective 768^3 resolution and used approximately 1 million CPU-hours. (Right) CASTRO simulation of nucleosynthesis during the explosion phase of a SN Ia. Shown are the nuclear burning products (orange = iron, light blue = silicon and calcium, dark blue = helium). This simulation was performed by Haitao Ma at UC Santa Cruz using Franklin at NERSC with an effective 8192^3 resolution and used approximately 2 million CPU-hours.

2. Software Infrastructure

MAESTRO and CASTRO are implemented using the BoxLib framework developed in the Center for Computational Sciences and Engineering at LBNL. BoxLib is a hybrid C++/Fortran90 software system that provides support for the development of parallel structured-grid AMR applications. In BoxLib, the memory management, flow control, parallel communication, and I/O are abstracted from the physics-specific routines, thus enabling many different applications to be built on the same software framework. SEDONA is implemented in a modern C++ framework that supports the massively parallel Monte Carlo approach.

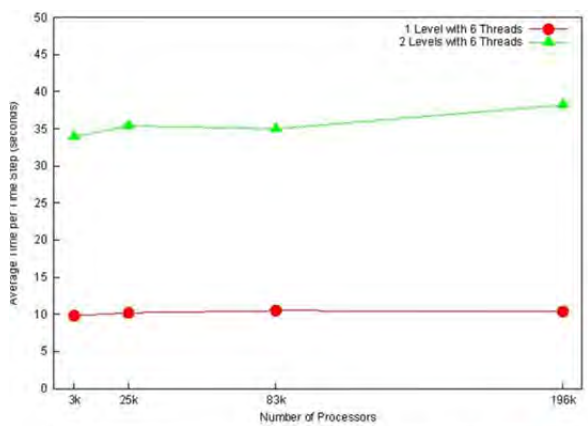
The fundamental parallel abstraction in BoxLib is the MultiFAB, which holds the data on the union of disjoint rectangular grids at a level of refinement. A MultiFAB is composed of FABs; each FAB is an array of data on a single grid. We use a coarse-grain parallelization strategy to distribute FABs to nodes, where the nodes communicate with each other using MPI. We also use a fine-grain parallelization strategy in the physics-based modules and the linear solvers, in which we use OpenMP to spawn a thread on each core on a node. Each thread operates on a portion of the associated FAB. FABs at each level of refinement are distributed independently.

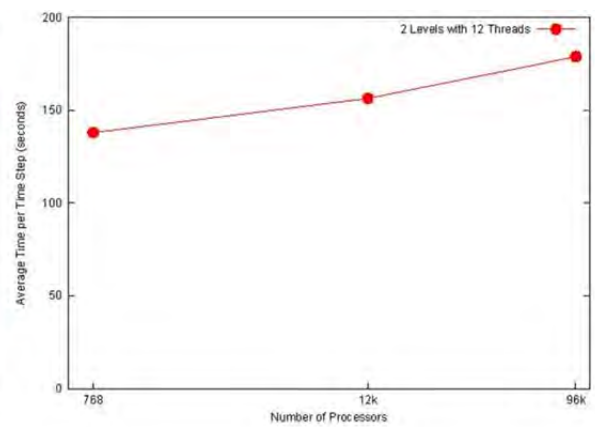
Each node contains meta-data that is needed to fully specify the geometry and node assignments of the FABs. At a minimum, this requires the storage of an array of boxes specifying the index space region for each AMR level of refinement. The meta-data can thus be used to dynamically evaluate the necessary communication patterns for sharing data between nodes for operations such as filling data in ghost cells and synchronizing the solution at different levels of refinement. Evaluating these communication patterns requires computation of the intersections of the grids themselves with rectangular patches that represent grids with ghost cells. A simple, brute force algorithm for doing so requires $O(N^2)$ operations, where N is the number of grids. This operation becomes expensive for problems with large numbers of grids, so we have implemented a hash sorting algorithm to reduce the

cost. Essentially, we subdivide the domain into multiple rectangular regions of index space, and sort the grids into these regions based on the lowest value in index space of each grid. Each region is large enough that a grid based in one region extends no further than the nearest neighbor regions. We use the knowledge of which region each grid "lives in" to restrict our search for intersecting grids to only that region and its neighbors. If M is the number of regions covering the domain, this reduces an $O(N^2)$ operation to an $O(N + N^2 / M)$ operation. In order to reduce the number of times the hash sort is called, we cache communication patterns that are most frequently used.

3. Scaling Results

We present scaling results demonstrating that our codes can efficiently run on the largest supercomputers (see Figure 2). We use a weak scaling approach, in which the number of cores increases by the same factor as the number of unknowns in the problem. For the MAESTRO runs, we keep the one-dimensional radial base state fixed in time for this study; for the CASTRO runs we use the monopole approximation for self-gravity. In the MAESTRO and CASTRO tests, we simulate a full star on a three-dimensional grid. In the multilevel calculations the inner 12.5% of the domain is refined. The results were obtained using Jaguar at OLCF, in which two hex-core sockets share memory on a node. Thus we either assign one MPI process per socket (in which case we spawn 6 threads), or one MPI process per node (in which case we spawn 12 threads). In each case, a single thread is assigned to a single core. We note that CASTRO scaling behavior is relatively insensitive to using 6 or 12 threads. MAESTRO has better scaling performance when using 12 threads at a cost of additional thread overhead time due to threading across different sockets. The SEDONA scaling test was performed using Intrepid at ANL using a pure-MPI approach, and shows the parallel performance expected of a Monte Carlo method.





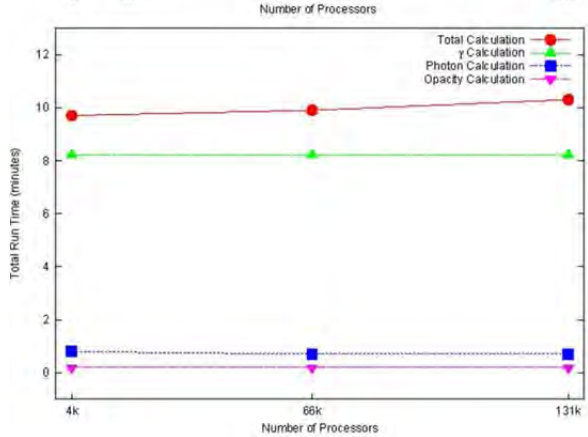


Figure 2. (Top-Left) CASTRO scaling results and (Top-Right) MAESTRO scaling results on Jaguar at OLCF. (Bottom) SEDONA scaling results using Intrepid at ANL. Each test uses a weak scaling approach in which the number of cores increases by the same factor as the number of unknowns in the problem. With perfect scaling the curves would be flat.

As shown in Figure 2, CASTRO scales well for the single-level and multilevel problems. We can also determine the AMR overhead using this data. Because of subcycling in time, a coarse time step consists of a single step on the coarse grid and two steps on the fine grid. Thus, we would expect that the time to advance the multilevel solution by one coarse time step would be a factor of three greater than the time to advance the single-level coarse solution by one coarse time step, plus any additional overhead associated with AMR. From the data in the figure we conclude that AMR introduces a modest overhead, ranging from approximately 15% for the 4,000 core case to 18% for the 196,000 core case. By contrast, advancing a single-level calculation at the finer resolution by the same total time, i.e., two fine time steps, would require a factor of 16 more resources than advancing the coarse single-level solution.

The overall scaling behavior for MAESTRO is not as close to ideal as that of CASTRO due to the linear solves performed at each time step. However, MAESTRO is able to take a much larger time step than CASTRO for flows in which the velocity is a fraction of the speed of sound, enabling the longer integration times needed to study convection.

4. End-to-End Capability

Performing an end-to-end simulation requires that a CASTRO simulation be initialized with the correctly transformed data from a MAESTRO simulation, and that SEDONA be initialized with data from a CASTRO simulation. SEDONA takes as input the density, velocity and compositional structure of the material ejected in the explosion and synthesizes emergent model spectra, light curves and polarization, which can then be compared directly against observations. This stage of the end-to-end simulation capability is straightforward; the only remaining task is to modify SEDONA to read the data from CASTRO's AMR hierarchy rather than from a uniform grid.

Initializing a CASTRO simulation with data from a MAESTRO simulation is analytically more complicated due to the difference between the low Mach number approach and a fully compressible approach. However, the fact that MAESTRO and CASTRO share a common software framework makes the implementation straightforward. Here we demonstrate the successful mapping from MAESTRO to CASTRO for a two-dimensional test problem, that of an in flowing jet.

The computational domain is 1 cm on each side, and the pressure and density are set to terrestrial conditions with zero initial velocity. At the inflow face, we apply a normal velocity with a maximum Mach number of 0.1, specifically,

$$v = c_s \{0.01 + 0.045[\tanh(100(x - 0.4)) + \tanh(100(0.6 - x))]\} \text{ cm} = \text{s} \quad . \tag{1}$$

The inflow density is set to half of the initial value inside the domain. In Figure 3, we show the density and pressure fields computed with MAESTRO and CASTRO to t = 300 μs. In the CASTRO simulation, an acoustic wave is launched from the inflow boundary. The acoustic signal bounces around the domain until later times, when the solution has mostly equilibrated. In Figure 4, we show the results from initializing a CASTRO simulation using the MAESTRO data from t = 200 μs. Shortly afterwards, the acoustic signal originating from the inflow boundary has equilibrated, and the final-time data closely matches the simulations in Figure 3.

Acknowledgements

We would like to thank Gunther Weber and Hank Childs of LBNL for their help in using the VisIt visualization software. We thank Ken Chen, Candace Joggerst, Haitao Ma, and Jason Nordhaus for being patient early users of CASTRO, and Chris Malone for his early work with MAESTRO. The work at LBNL was supported by the SciDAC Program of the DOE Office of Mathematics, Information, and Computational Sciences under the U.S. Department of Energy under contract No. DE-AC02-05CH11231. The work at Stony Brook was supported by a DOE/Office of Nuclear Physics Outstanding Junior Investigator award, grant No. DEFG02-06ER41448, to Stony Brook.

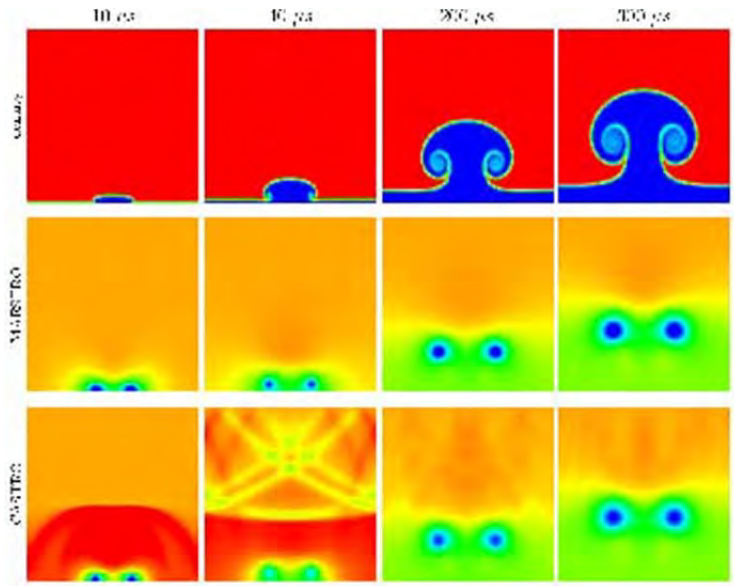


Figure 3. Evolution of a low Mach number jet showing (top) density, (middle) MAESTRO pressure, (bottom) CASTRO pressure for the inflow jet problem at 4 different times in the evolution. The density plots are indistinguishable between the MAESTRO and CASTRO simulations.

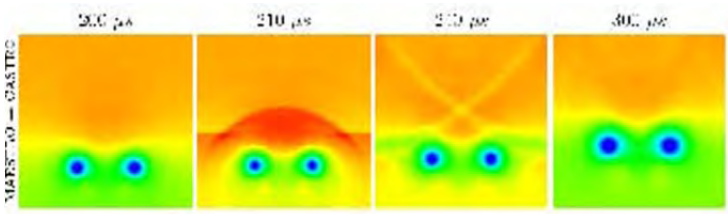


Figure 4. Jet evolution using the MAESTRO dataset from t = 200 μs (see Figure 3) to initialize a CASTRO simulation. Here, the time sequence corresponds to the last two columns of Figure 3.

References

[1] A. S. Almgren, V. E. Beckner, J. B. Bell, M. S. Day, L. H. Howell, C. C. Joggerst, M. J. Lijewski, A. Nonaka, M. Singer, and M. Zingale. CASTRO: A new compressible astrophysical solver. I. Hydrodynamics and self-gravity. *The Astrophysical Journal*, 215:1221–1238, 2010.

[2] D. Kasen, R. C. Thomas, and P. Nugent. Time-dependent Monte Carlo radiative transfer calculations for three-dimensional supernova spectra, light curves, and polarization. *The Astrophysical Journal*, 651:366–380, November 2006.

[3] A. Nonaka, A. S. Almgren, J. B. Bell, M. J Lijewski, C. M Malone, and M. Zingale. MAESTRO: An adaptive low mach number hydrodynamics algorithm for stellar flows. *The Astrophysical Journal Supplement Series*, 188:358–383, 2010.

Simulation of nitrogen emissions in a low swirl burner

J. B. Bell, M. S. Day, X. Gao, M. J. Lijewski

Center for Computational Science and Engineering
Lawrence Berkeley National Laboratory

Abstract. We present simulations of a laboratory-scale low swirl burner fueled with hydrogen at atmospheric pressure. The study uses detailed chemistry and transport including a sub-mechanism for computing nitrogen emissions. The simulation shows how the cellular burning structures characteristic of lean premixed hydrogen combustion lead to enhancements in the NOx emissions from these flames. Analysis of the simulation data illustrates the chemical pathways that lead to nitrogen emissions and how they are enhanced within local regions of intense burning.

1. Introduction

Combustion devices based on lean premixed flames generate substantially lower NO_x emissions than equivalent nonpremixed or stoichiometric premixed systems [3]. For lean premixed systems, the reduction in NO_x emissions is believed to be a direct consequence of the lower post-flame gas temperatures. Recent laboratory experiments have shown that ultra-lean hydrocarbon flames (with fuel concentrations well below lean blow-off limits) can in fact be stabilized in a turbulent burner if the fuel mixture is enriched with H_2 [8]. Hydrogen addition enhances the effective flame speed and auto-ignition processes of the mixture [7], both of which lead to an increased resistance to strain-induced extinction. However, NO_x emission do not decrease uniformly with increasing hydrogen enrichment relative to the hydrocarbon fuel. In a recent experimental study [2], pure H_2-air flames at 8 atmospheres exhibited a pronounced "floor" at 1 ppm NO in the exhaust stream, below which further reduction in the adiabatic post-flame gas temperature had no effect on the measured emissions. Moreover, this "floor" was a factor of two or three higher than values produced in the same burner using ultra-lean hydrocarbon mixtures (seeded with hydrogen) that had comparable post-flame gas temperatures.

Here, we simulate a laboratory scale model of a low swirl burner fueled by hydrogen at an equivalence ratio of $\phi = 0{:}37$ at atmospheric pressure with nitrogen emissions chemistry. Our goal is to identify the underlying mechanism that control the formation of nitrogen emissions in this type of flame. To accurately capture the emissions formation and dynamics, simulations are performed with a detailed kinetics model for hydrogen chemistry including the relevant nitrogen chemistry and detailed transport. The simulations fully resolve the turbulent flow and internal flame structure so that no model for turbulence/chemistry is required.

2. Numerical Methodology

The simulations presented here are based on a low Mach number formulation [13,9] of the reacting Navier-Stokes equations. The methodology treats the fluid as a mixture of perfect gases ignoring Soret, Dufour and radiative transport processes. We use a mixture-averaged model for differential species diffusion, which is critical for capturing the thermodiffusive behavior of lean hydrogen flames

(see [6] for a complete discussion of this approximation). A detailed chemical description containing 15 species and 58 reversible Arrhenius reactions is used to describe premixed hydrogen-air combustion and subsequent NO_x formation. This mechanism was derived by rate-of-production and sensitivity analyses from a full reaction mechanism [10,11] to obtain the minimum reaction set that describes NO formation in H_2 oxidation at lean conditions. It includes NO formation routes through NNH and N_2O, as well as the thermal mechanism. The reactions involving nitrogen emission are presented in Table 1. Addition detail along with two dimensional studies of NO_x emissions in H_2 flames at various equivalence ratios are presented in [1].

Table 1. Thermal and NNH/N_2O formation routes for NO. For each reaction, i, the forward progress rate $k_i = A_i T^{n_i} \exp(-E_i/RT)$ ($moles/cm^3 \cdot s$), where A_i, n_i, E_i, ($cal/g \cdot K$) are given below.

No.	Reaction	A	n	E
30	$NO+O(+M) \rightleftharpoons NO_2(+M)$	1.3E15	-0.75	0
31	$NO+HO_2 \rightleftharpoons NO_2+OH$	2.1E12	0.00	-497
32	$NO_2+H \rightleftharpoons NO+OH$	1.3E14	0.00	362
33	$NO_2+O \rightleftharpoons NO+O_2$	1.1E14	-0.52	0
34	$2\ NO_2 \rightleftharpoons 2\ NO+O_2$	4.5E12	0.00	27599
35	$N_2O(+M) \rightleftharpoons N_2+O(+M)$	1.3E12	0.00	62570
36	$N_2O+H \rightleftharpoons N_2+OH$	3.3E10	0.00	4729
37	$N_2O+H \rightleftharpoons N_2+OH$	4.4E14	0.00	19254
38	$N_2O+O \rightleftharpoons NO+NO$	9.2E13	0.00	27679
39	$N_2O+O \rightleftharpoons N_2+O_2$	3.7E12	0.00	15936
40	$NH+H \rightleftharpoons N+H_2$	3.0E13	0.00	0
41	$NH+O \rightleftharpoons NO+H$	9.2E13	0.00	0
42	$NH+OH \rightleftharpoons N+H_2O$	5.0E11	0.50	2000
43	$NH+O_2 \rightleftharpoons NO+OH$	1.3E06	1.50	100
44	$NH+NO \rightleftharpoons N_2O+H$	2.9E14	-0.40	0
45	$NH+NO \rightleftharpoons N_2O+H$	-2.2E13	-0.23	0
46	$NH+NO \rightleftharpoons N_2+OH$	2.2E13	-0.23	0
47	$NH+NO_2 \rightleftharpoons N_2O+OH$	1.0E13	0.00	0
48	$N+OH \rightleftharpoons NO+H$	3.8E13	0.00	0
49	$N+O_2 \rightleftharpoons NO+O$	6.4E09	1.00	6280
50	$N+NO \rightleftharpoons N_2+O$	2.1E13	0.00	0
51	$NNH \rightleftharpoons N_2+H$	6.5E07	0.00	0
52	$NNH+H \rightleftharpoons N_2+H_2$	1.0E14	0.00	0
53	$NNH+O \rightleftharpoons N_2O+H$	1.0E14	0.00	0
54	$NNH+O \rightleftharpoons N_2+OH$	8.0E13	0.00	0
55	$NNH+O \rightleftharpoons NH+NO$	5.0E13	0.00	0
56	$NNH+OH \rightleftharpoons N_2+H_2O$	5.0E13	0.00	0
57	$NNH+O_2 \rightleftharpoons N_2+HO_2$	2.0E14	0.00	0
58	$NNH+O_2 \rightleftharpoons N_2+H+O_2$	5.0E13	0.00	0

The numerical methodology uses a finite-volume numerical discretization [4] that combines a symmetric operator-split treatment of chemistry and transport with a density-weighted approximate projection method to impose the low Mach number constraint. The integration proceeds on the time scale of the advective transport; faster diffusion and chemistry processes are treated implicitly in time. The integration scheme is embedded in an adaptive mesh refinement algorithm based on a hierarchical system of rectangular grid patches. The complete integration algorithm is second-order accurate in space and time, and discretely conserves species mass and enthalpy. The methodology is implemented in parallel using a hybrid programming model. Patches of data are distributed over nodes using a coarse-grained load distribution strategy based on MPI. Within nodes, fine-grained parallelization within patches is implemented using a thread-based approach using OpenMP. The hybrid implementation scales well to 50K processors.

The configuration uses a 2.5 cm radius nozzle with mean flow specified from experimental measurements provided by Petersson et al. [12] for a case with mean flow of 6 m/s, scaled uniformly by 250% to approximate profiles at a higher fueling rate of 15 m/s. Turbulent fluctuations matched to experimentally measured turbulent intensities and integral length scales are added to the mean flow. Outside the nozzle, a 35 cm/s upward coflow of air at 300K is imposed.

The simulation domain measures 25 cm × 250 cm × 25 cm and is discretized using a base mesh of 256^3. Three levels of grid refinement, each by a factor of 2, dynamically track regions of high vorticity (turbulence) and reactivity (combustion/flame), and the resulting effective resolution is 2048^3. Note that this configuration, and the simulation parameters, are identical to the high-turbulence lean hydrogen case discussed in [5], except for the inclusion of detailed emission chemistry. In that work, we demonstrated that this level of resolution is adequate to capture the detailed structure of the flame including the peak fuel consumption, the thermal field and major species.

3. Simulations Result and Analysis

The inclusion of emissions chemistry does not change the basic flame structure, which is described in [5]. The overall flame shape and structure of the NO profile are shown in Figure 1. Here, we focus on the details of the nitrogen chemistry in the core of the flame above the nozzle, specifically for $r < 2.5$ cm. For this analysis, we construct a local coordinate system around the flame using T as a progress variable. We identify the flame surface with the $T = 1144$ isotherm and extend the coordinate system on either side by following integral curves of ∇T. We then construct a triangularization of the original surface. For each triangle, we extend off the surface to form tubes with triangular cross-section which we will refer to as "flamelets." We then sample T, fuel consumption and reaction rates associated with the nitrogen chemistry over each flamelet, which are then sorted and binned according to the peak fuel consumption rate, and conditionally averaged in temperature space.

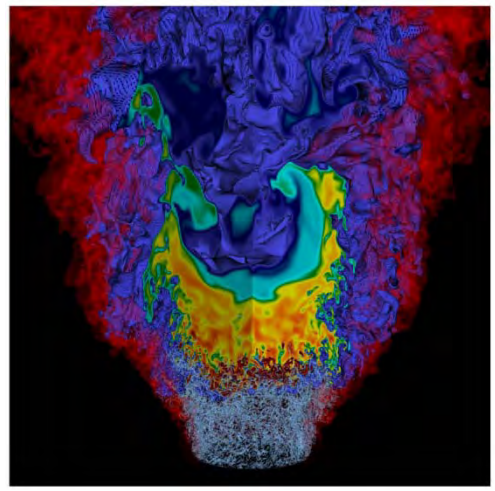


Figure 1. Cutaway of flame snapshot. Isovolume (enclosed in blue, colored on exposed surfaces) of NO concentration, fuel boundary is semi-opaque red surface, and grey represents vertical structures at shear layer of turbulent swirling flow.

Figure 2(a) shows the distribution of burning strength bins where they intersect the T = 1144K isotherm. For several representative bins, Figures 2(b–d) show conditional mean NO production rates with temperature for the dominant contributors using the numbering in Table 1. The conditional mean total NO production rate (shown as the solid black line in the Figure 2) consists primarily of contributions from the "thermal" ($\omega_{NO-.therm}$, dotted black line) and NNH (ω_{NO-NNH} dashed black line) routes. The thermal route consists of three reactions that create NO from O_2, N_2, and the N radical; these are typically associated with high temperature ($T \gtrsim 1700$). The NNH route is initiated when N_2 combines with H to form NNH. NNH then reacts to produce either N_2O or the pair, NO and NH. Of the N_2O created, approximately 90% is converted back to N_2, and the remainder reacts to NO. The NH reacts to N, and then proceeds to NO through the thermal route. Even though most of the nitrogen passing through the NNH route is recycled back to N_2, the fraction that reacts to NO still accounts the bulk of the NO production. It is interesting that both the NNH and thermal rates become significant when $T \gtrsim 1300$; the adiabatic flame temperature for this fuel mixture is 1360 K, and peak temperatures are observed in excess of 1740 K across all flamelet groups. Similar increases in NO production by the NNH and thermal routes were observed in methane-air mixtures seeded with H_2 [7].

Figure 2 suggests that although the strongest burning regions produce NO at a rate that is as high as 20 times that of the weakly burning regions at comparable temperatures, most of the flame is actually burning at relatively weak rates. Figure 3(a) shows integrated NO production as a function of T, showing that the majority of NO is generated near 1500 K. Also, we see that ≈80% of the NO is created through the NNH route, while ≈20% is generated via the thermal route. Figure 3(b) shows a quantitative reaction path diagram, where the thickness of the graph edges is scaled to the transfer rate of nitrogen atoms between species pairs. The data is integrated over the flame region where $r < 2.5$cm, and over the entire temperature range. Contributions to each edge are collected into 3 burning groups (see caption), and represented as percentages of total NO production. The $N_2 \rightarrow N_2O$ path shows a negative contribution for the weak flames, indicating that the path actually reverses. In spite of the extremely large NO production rates in the hotspot regions, the weakly burning parts of the flame account for over half of the total NO production.

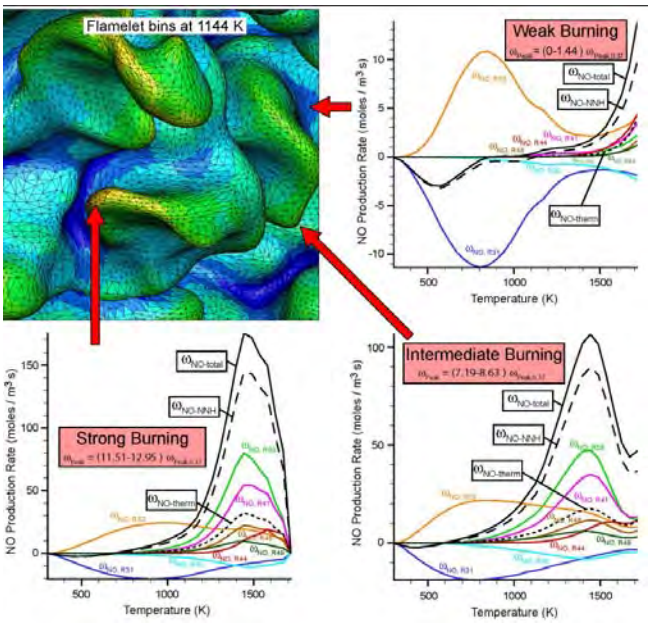


Figure 2. Emission chemistry in a turbulent lean hydrogen-air low-swirl flame. The flame surface is divided into local flamelets, and binned by peak burning rate. (a) T = 1144K, colored where it intersects the flamelets by corresponding bin number (blue = 1, red = 10). (b–d) Conditional mean rates for the dominant reactions producing NO, along with the total NO production (black curve). Dotted black curves represent the total contribution due to the "thermal" sub-mechanism; dashed curves indicate the total due to the NNH sub-mechanism.

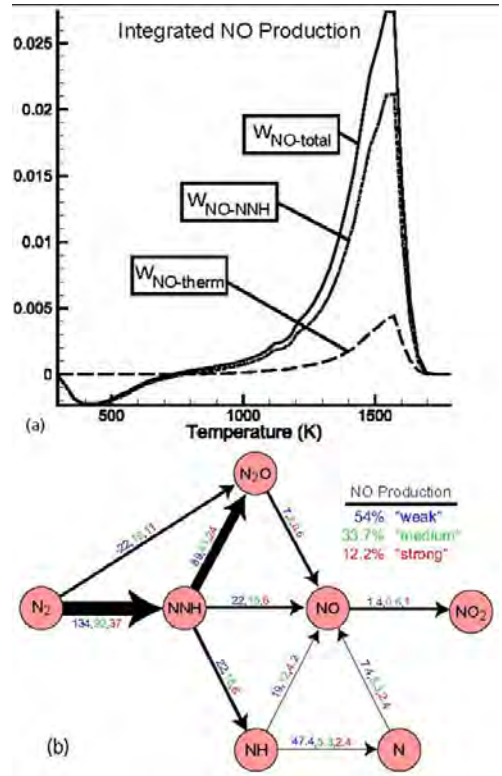


Figure 3. (a) Distribution of integrated emissions production and (b) Path diagram quantifying the integrated transfer rate of N atoms between species, normalized by the total rate of N atoms into NO. Each path indicates the contribution (in percent of NO production rate) from the weak (ω_{Peak} = 0 – 4$\omega_{Peak,0.37}$, blue), intermediate (ω_{Peak} = 4 – 9$\omega_{Peak,0.37}$, green) and strong (ω_{Peak} = 9 – 13$\omega_{Peak,0.37}$, red) flames, where $\omega_{Peak,0.37}$ is the peak consumption rate of the flat flame at ϕ = 0.37.

The premixed hydrogen flame simulated here shows the cellular structure characteristic of thermodiffusively unstable flames. These cellular structures lead to significant enhancement in the local burning and, not surprisingly, increases in the level of nitrogen emissions. Local flamelet analysis confirms this behavior and that the dominant pathway is through NNH. Similar results were shown in [1]; however, the more intense turbulence here leads to increased emissions.

Future work will examine the emissions chemistry in greater detail including NO_2 formation and emissions characteristics in the shear-layer region of the nozzle flow. Finally, we will look at the effects of elevated pressure and a range of equivalence ratios in order to confirm that the processes we have identified are applicable to the experimental data showing the NO_x floor.

References

[1] J. B. Bell, M. S. Day, X. Gao, and P. Glarborg. Numerical simulation of nitrogen oxide formation in lean premixed turbulent $H_2/O_2/N_2$ flames. *Proc. Combust. Inst.*, 33:to appear, 2010.

[2] R. K. Cheng, D. Littlejohn, P. A. Strakey, and T. Sidwell. Laboratory investigation of a low-swirl injector with H_2 and CH_4 at gas turbine conditions. *Proc. Combust. Inst.*, 32:3001–3009, 2009.

[3] S. M. Correa. A review of NO_x formation under gas-turbine combustion conditions. *Combust. Sci. Technol.*, 87:329–362, 1992.

[4] M. S. Day and J. B. Bell. Numerical simulation of laminar reacting flows with complex chemistry. *Combust. Theory Modelling*, 4:535–556, 2000.

[5] M. S. Day, J. B. Bell, R. K. Cheng, S. Tachibana, V. E. Beckner, and M. J. Lijewski. *Cellular burning in lean premixed turbulent hydrogen-air flames: coupling experimental and computational analysis at the laboratory scale, volume 180 of Journal of Physics Conference Series: SciDAC 2009* (W. Tang, Ed.). Institute of Physics Publishing, San Diego, CA, 2009.

[6] A. Ern and V. Giovangigli. *Multicomponent Transport Algorithms, volume m24 of Lecture Notes in Physics*. Springer-Verlag, Berlin, 1994.

[7] E. R. Hawkes and J. H. Chen. Direct numerical simulation of hydrogen-enriched lean premixed methane-air flames. *Combust. Flame*, 138:242–258, 2004.

[8] D. Littlejohn and R. K. Cheng. Fuel effects on a low-swirl injector for lean premixed gas turbines. *Proc. Combust. Inst.*, 31:3155–3162, 2007.

[9] A. Majda and J. A. Sethian. The derivation and numerical solution of the equations for zero Mach number combustion. *Combust. Sci. Technol.*, 42:185–205, 1985.

[10] T. Mendiara and P. Glarborg. Ammonia chemistry in oxy-fuel combustion of methane. *Combust. Flame*, 156:1937–1949, 2009.

[11] T. Mendiara and P. Glarborg. Reburn chemistry in oxy-fuel combustion of methane. *Energ. Fuels*, 23:3565–3572, 2009.

[12] P. Petersson, J. Olofsson, C. Brackman, H. Seyfried, J. Zetterberg, M. Richter, M. Alden, M. Linne, R. Cheng, A. Nauert, D. Geyer, and A. Dreizler. Simultaneous PIV/OH PLIF, Rayleigh thermometry/OH PLIF and stereo PIV measurements in a low-swirl flame. *Appl. Opt*, 46:3928–3936, 2007.

[13] R. G. Rehm and H. R. Baum. The equations of motion for thermally driven buoyant flows. *N. B. S. J. Res.*, 83:297–308, 1978.

Algorithms for Lattice Field Theory at Extreme Scales.

Richard C. Brower[1], Ronald Babich[1], James Brannick[2], Michael A. Clark[3], Saul Cohen[1], Balint Joo[4], Anthony Kennedy[5], James Osborn[6], Claudio Rebbi[1]

[1]Center for Computational Science, Boston University, Boston, MA 02215, USA
[2]Department of Mathematics, Penn State University, University Park, PA 16802, USA
[3]Harvard-Smithsonian Center for Astrophysics, Cambridge, MA 02138, USA
[4]Scientific Computing Group, Jefferson Lab, Newport News, VA 23606, USA
[5]School of Physics & Astronomy, University of Edinburgh, Edinburgh EH9 3JZ, UK
[6]Argonne Leadership Computing Facility, Argonne National Lab., Argonne, IL 60439, USA

E-mail: `brower@bu.edu`

Abstract. Increases in computational power allow lattice field theories to resolve smaller scales, but to realize the full benefit to scientific discovery, new multi-scale algorithms must be developed to maximize efficiency. Examples of new trends in algorithms include adaptive multigrid solvers for the quark propagator [1] and an improved symplectic Force Gradient integrator [2] for the Hamiltonian evolution used to include the quark contribution to vacuum fluctuations in the quantum path integral. Future challenges to algorithms and software infrastructure targeting many-core GPU accelerators [3] and heterogeneous extreme scale computing are discussed.

1. Introduction

Lattice field theory (LFT) provides a framework for the numerical solution of Quantum Chromodynamics (QCD), the fundamental theory for Nuclear Physics (NP) and the crucial non-perturbative core of the Standard Model for High Energy Particle Physics (HEP). Remarkable progress has been made in recent years by the development of a range of algorithmic and software tools and advanced high performance architectures supported in large part by SciDAC and the INCITE programs at the Deparment of Energy. This has enabled *ab initio* calculations of important parameters at a precision comparable to and sometimes exceeding experimental results. A new application of lattice field theory is the exploration of possible mechanisms beyond the standard model for the dynamical breaking of electro-weak symmetry, the current focus of the experimental program at the Large Hadron Collider (LHC) in Geneva.

These advances in lattice QCD simulation have been largely based on a *simple* but clearly successful single scale algorithmic and software paradigm. The fundamental data structure of lattice field theory is a uniform hypercubic grid. On this grid we require two basic operations:

- A sparse matrix solver to compute the propagation of fermions (e.g.quarks) in a background of "rough" chromo-electromagnetic (gluon) fields.
- A symplectic integrator for the Hamiltonian evolution to produce large ensembles of typical background field configurations.

Efficient algorithms for both operations exploit the simplicity of a uniform and static map from the 4-d lattice onto a homogeneous parallel architecture. Each processing element is assigned a regular space-time subvolume communicating with its neighbors on the 4-d hypercubic grid. This approach coupled with steady improvements of the Krylov solvers and symplectic integrators has indeed yielded impressive results. **However the underlying single scale paradigm of lattice field theory software will almost certainly fail to sustain optimal performance at extreme scales.**

The reasons for this paradigm shift are twofold: physics and hardware. First as the power of computational resources increases, accuracy is gained by resolving smaller length scales. This means going to larger lattice volumes (L_{box}) and smaller lattice spacing ($a_{lattice}$). As a result traditional local relaxaation algorithms become ill-conditioned and suffer from critical slowing down. To take full advantage of this increased power requires new multi-scale algorithmic advances. Second it appears inevitable that extreme scale computer architectures will become increasingly heterogeneous, composed of nodes with O(1000) cores integrated into a hierarchical network. There is an urgent need to focus resources on the multi-scale/many-threaded restructuring of the lattice field theory software infrastructure required in the transition to the exascale. To co-ordinate this transformation calls for co-design spanning the disciplines from physics, to applied mathematics, to computer science and computer architecture.

2. Multiple scales in Lattice Field Theory

Lattice QCD is intrinsically multi-scaled. This is not obvious from the Lagrangian (or classical differential equations) as the only scale is the quark mass, which to a tolerable accuracy can be set to zero. However the mysterious property of quark confinement introduces an intrinsic scale[1] ($l_\sigma \simeq m_{proton}^{-1}$) reflected clearly in the substructure of quantum fluctuations of the chromo-electromagnetic field illustrated [4] in Fig. 1. This scale is a result of so called "spontaneous breaking of conformal or scale invariance". Then another non-perturbative phenomena, "spontaneous breaking of chiral symmetry", gives a larger length controlled by the two lightest quark masses. For QCD this accounts for a scale hierarchy for the pion relative to the proton of $m_{proton}/m_\pi \simeq 7$. Combined with the artificial numerical scales of any computer simulations: the lattice spacing ($a_{lattice}$) and the box size (L_{box}), we have a multi-scale hierarchy,

$$a_{lattice} \ll l_\sigma \simeq m_{proton}^{-1} \ll m_\pi^{-1} \ll L_{box} \ , \tag{1}$$

[1] In natural units with $c = \hbar = 1$, length scales are the same as inverse masses.

that current advances in hardware and algorithms are just beginning to fully accommodate. To accomodate this hierarchy requires lattice volumes on order of $L^4 = 100^4$ just beginning to be practicable on Petascale hardware. This is just one example of additional scales encountered in

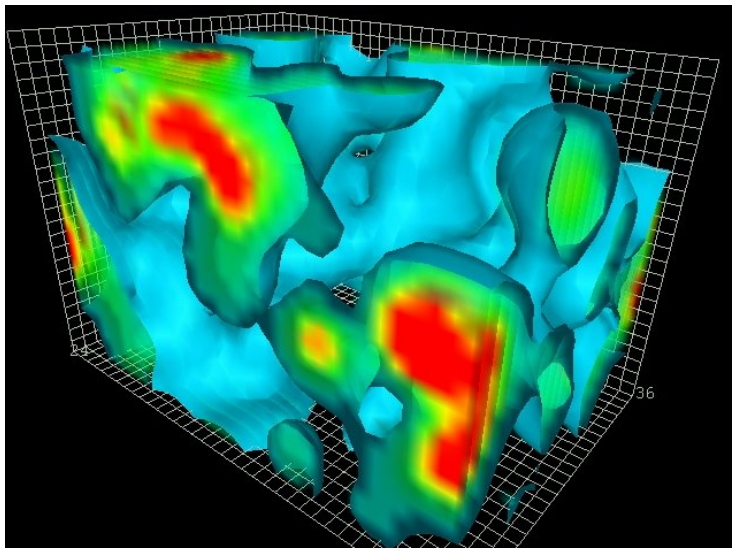


Figure 1. A typical fixed "time" slice of the four-dimensional space-time structure of gluon-field configurations contributing to the the lattice field theory ensemble [4].

lattice field theory applications; others include the quark gluon plasma at finite temperature, the binding of nucleons in nuclear physics, heavy quarks decays and strong binding alternatives to the Higgs mechanism, all of which will be of increasing interest as Exaflops computing enables new avenues of research. Already as we describe next, as a consequence of this scale heiarchy, we are beginning to seeing a clear advantage to multi-scale solvers relative to the best implementation of the conjugate gradient algorithm.

3. Adaptive Multigrid Dirac Solver

Multi-scale algorithms for QCD are still in their infancy. However after initial attempts almost 25 years ago the first successful QCD multigrid algorithm has finally been formulated[1]. The basic idea of using a coarse representation of the Dirac operator on lattices of increasing lattice spacing appears at first to be an obvious extension of the basic principles central to the renormalization group itself. Indeed early attempts, generally inspired by this observation, did succeed in formulating a variety of gauge invariant coarsening schemes but they all failed to improve convergence at the length scale l_σ, where the underlying lattice gauge field becomes rough. Indeed as Brower, Edwards, Rebbi and Vicari [5] demonstrated this failure occurred uniformly when the product of the mass gap m_q and the coherence length l_σ is of order one: $m_q l_\sigma = O(1)$.

The new element in Adaptive Multigrid for QCD [1] is the use of the slow convergence of the fine level solver to automatically discover the near null space and subsequently used these modes as interpolation elements to construct the coarse grid Dirac operator on the next level (see

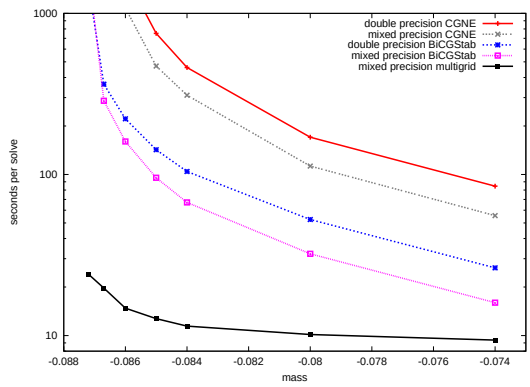


Figure 2. The adaptive multigrid performance for the Wilson-clover Dirac lattice operator as a function of quark mass relative to the best Krylov solvers.

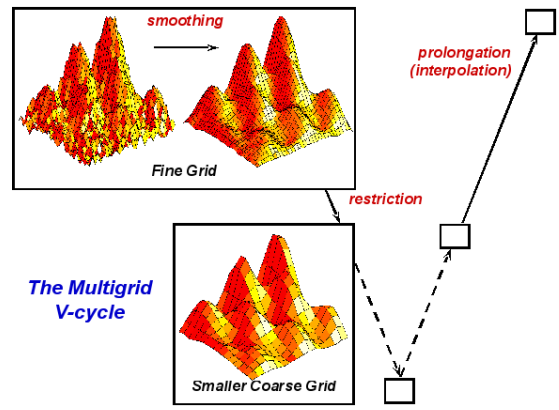


Figure 3. The "smooth" near null subspace is projected onto a coarse operator in a multigrid V-cycle.

Fig. 3). Applied to the Wilson-clover inverter on a $32^3 \times 256$ anisotropic lattice, it outperforms the best single scale methods by 20x at the lightest quark masses as demonstrated in Fig. 2. Extensions of adaptive multigrid are underway for Domain Wall and Staggered fermions as well as to Hamiltonian evolution to generate lattice ensembles.

4. Force Gradient Integrator

The ensemble of gauge configurations is generated by a Markov process for the equilibrium distribution of gluons interacting with the vacuum fluctuation of the fermion (e.g. quark) field. Because of the non-local contribution of the determinant of the Dirac matrix, this is accomplished using a so-called Hybrid Monte Carlo (HMC) process which requires the integration of Hamiltonian dynamics in "Monte Carlo time" followed by a Metropolis acceptance test. The optimization of the integration and acceptance involves splitting the Hamiltonian onto multiple time scales and employing a symplectic nested integration scheme. Current integrators split the Hamiltonian into separate terms with variable time steps, and a 4th order multi-time step symplectic integrator is used for each component. The Force Gradient integrator is a recent refinemnet that involves tuning the higher order contribution from the expansion in Poisson brackets of the exactly conserved "shadow Hamiltonian" to minimize the error relative to the continuum Hamiltonian [2]. As seen in the Fig. 4 below the step size can be increased at 75% acceptance rate relative to earlier symplectic integrators. This improves the efficiency substantially, particularly in the limit of light quark masses.

5. GPU Testbed and Future Directions

The exascale era promises to dramatically expand the ability of lattice field theory to investigate the multiple scales in nuclear physics, the quark-gluon plasma as well as possible strong dynamics

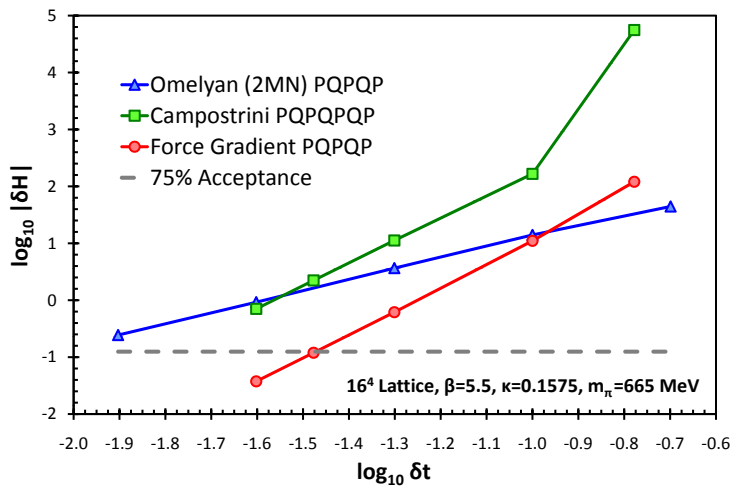


Figure 4. Change δH in the Hamiltonian over an entire trajectory as a function of the integration step size $\delta\tau$. The dashed line indicates the value of δH that corresponds to a 75% HMC acceptance rate.

beyond the standard model. Increasingly complex scale-aware QCD algorithms are a challenge to software engineering and the co-design of the heterogeneous architecture to support them. At present the multi-GPU cluster offers a useful preview of the challenge at the level of 100's of cores per node with a relatively low bandwidth interconnect. Development of new algorithms to meet this challenging architecture include communication reduction by (Schwarz) domain decomposition, multi-precision arithmetic, data compression and local reconstruction. The QUDA library for QCD on GPU clusters[3] being developed at Boston University under the SciDAC project provides a software platform for these investigations. As an early hint of the marriage of multi-scale algorithms with many-core technologies, we estimate that the 20x speed up of the multigrid solver above when combined with 5x cost improvement of the current Nvidia Fermi processor will give a multiplicative reduction of the cost per Dirac solve of $O(100)$. While very promissing, we note that extending this initial multi-scale methodology to full scale lattice calculation remains a very challenging problem at the intersection of physics, applied mathematics and computer science with tremendous potential for advancing this science.

6. References

[1] R. Babich, J. Branick, R. C. Brower, M. A. Clark, S.F. Manteuffel, S. F. McCormick, J. C. Osborn and C. Rebbi, "Adaptive multigrid algorithm for the lattice Wilson-Dirac operator", arXiv:1005.3043 [hep-lat].

[2] A. D. Kennedy, M. A. Clark and P. J. Silva, "Force Gradient Integrators," arXiv:0910.2950 [hep-lat].

[3] M. A. Clark, R. Babich, K. Barros, R. C. Brower and C. Rebbi, "Solving Lattice QCD systems of equations using mixed precision solvers on GPUs," arXiv:0911.3191 [hep-lat].

[4] Derek B.Leinweber, "Visualizations of Quantum Chromodynamics," http://www.physics.adelaide.edu.au/dleinweb/VisualQCD/Nobel.

[5] R. C. Brower, R. G. Edwards, C. Rebbi and E. Vicari, "Projective multigrid for Wilson fermions," Nucl. Phys. B **366**, 689 (1991).

Evaluating systematic dependencies of Type Ia supernovae

A. C. Calder,[1,2] B. K. Krueger,[1] A. P. Jackson,[1] D. M. Townsley,[3] F. X. Timmes,[4] E. F. Brown,[5] and D. A. Chamulak[6]

[1]Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794-3800
[2]New York Center for Computational Sciences, Stony Brook University, Stony Brook, NY 11794-3800
[3]Department of Physics and Astronomy The University of Alabama, Tuscaloosa, AL 35487-0324
[4]School of Earth and Space Exploration, Arizona State University, Tempe, AZ 85287-1404
[5]Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824-2320
[6]Physics Division, Argonne National Laboratory, Argonne, IL 60439

E-mail: acalder@mail.astro.sunysb.edu

Abstract. Type Ia supernovae are bright stellar explosions thought to occur when a thermonuclear runaway consumes roughly a solar mass of degenerate stellar material. These events produce and disseminate iron-peak elements, and properties of their light curves allow for standardization and subsequent use as cosmological distance indicators. The explosion mechanism of these events remains, however, only partially understood. Many models posit the explosion beginning with a deflagration born near the center of a white dwarf that has gained mass from a stellar companion. In order to match observations, models of this single-degenerate scenario typically invoke a subsequent transition of the (subsonic) deflagration to a (supersonic) detonation that rapidly consumes the star. We present an investigation into the systematics of thermonuclear supernovae assuming this paradigm. We utilize a statistical framework for a controlled study of two-dimensional simulations of these events from randomized initial conditions. We investigate the effect of the composition and thermal history of the progenitor on the radioactive yield, and thus brightness, of an event. Our results offer an explanation for some observed trends of mean brightness with properties of the host galaxy.

1. Introduction

Type Ia supernovae (SNe Ia) are bright stellar explosions distinguished principally by a lack of hydrogen and strong silicon features in their spectra (for reviews, see [1,2]). Properties of the light curves of these events allow their use as distance indicators at cosmological distances [3,4,5], and these are at present the most powerful and best proved technique for studying dark energy [6,7,8,9,10]. Accordingly, there are many observational campaigns underway striving to gather information about the systematics of these events and to better measure the expansion history of the Universe (see [11]

and references therein). These events are also responsible for producing many of the heavy (iron-group) elements found in the galaxy and are therefore critical to galactic chemical evolution [12].

Despite their widespread use as distance indicators and their importance to galactic chemical evolution, much of what is known about SNe Ia follows from empirical relationships, not a theoretical understanding of the explosion mechanism. Motivated principally by cosmological studies, observational campaigns are uncovering the rich phenomenology of these events at an unprecedented rate and the future promises even more (the Dark Energy Survey, LSST, JDEM, see [13]). Accordingly, the goal of modeling SNe Ia is a theoretical understanding of the observed properties, particularly the intrinsic scatter of these events and the source of any systematic trends. Understanding and quantifying these are essential to their effective use as distance indicators [5,14].

A widely accepted view is that SNe Ia are the thermonuclear incineration of a white dwarf composed principally of carbon and oxygen that has gained mass from a companion star [2]. In this scenario, the white dwarf gains mass, compressing the core until an explosion ensues. Recent observational evidence, however, does suggest other progenitors such as the merging of two white dwarf may explain many events [15,16,17]. Regardless of the exact mechanism, the "upshot" is that these events synthesize ~$0.6M_{\odot}$ of radioactive ^{56}Ni, and the decay of this radioactive ^{56}Ni, not the explosion energy, powers the light curve. There is a correlation, obeyed by the vast majority of SNe Ia, between the peak brightness and the timescale over which the light curve decays from its maximum. This "brighter is broader" relation [3] explains the property of these events that allows their use as distance indicators—by observing the decline from maximum light, one can infer the peak brightness. The correlation is understood physically as stemming from having both the luminosity and opacity being set by the mass of 56Ni synthesized in the explosion [18,19,20].

1.1. Observational trends of SNe Ia

Many contemporary observations address correlations between the event and properties of the progenitor galaxy. Of particular interest are correlations between the brightness of an event and the isotopic composition of a galaxy and its age, measured by the intensity of star formation.

The proportion of material that has previously been processed in stars, i.e., elements other than hydrogen and helium, which are collectively referred to as "metals," is a measurable property of the galaxy. (The relative abundance of these elements is referred to as the galaxy's "metallicity.") The presence of these elements in a progenitor white dwarf influences the outcome of the explosion by changing the path of nuclear burning, which influences the amount of ^{56}Ni synthesized in an event. Because the decay of ^{56}Ni powers the light curve, metallicity can directly influence the brightness of an event. Observational results to date are consistent with a shallow dependence of brightness on metallicity, with dimmer events in metal-rich galaxies, but are unable demonstrate a conclusive trend [21,22,23,24]. Determining the metallicity dependence is challenging because the effect appears to be small, is difficult to measure, and there are systematic effects associated with the mass-metallicity relationship within galaxies [22]. This effect is also difficult to decouple from the apparently stronger effect of the age of the parent stellar population on the mean brightness of SNe Ia [22,24,25].

When galaxies form, they are rich in hydrogen gas and undergo a period of intense star formation. Observations also target correlations between the brightness of an event and the age of a galaxy measured as the elapsed time from the period of intense star formation. Some observations indicate that the dependence of the SN Ia rate on delay time (elapsed time between star formation and the supernova event) is best fit by a bimodal distribution with a prompt component less than 1 Gyr after star formation and a tardy component several Gyr later [26,27]. Other studies only indicate a correlation between the delay time and brightness of SNe Ia with dimmer events occurring at longer delay times [28,24,23,29].

1.2. The deflagration to detonation transition model

We explore the systematics of these events with models that assume the explosion occurs in a carbon-oxygen white dwarf that has gained mass from a companion. Here we briefly describe this explosion scenario and the physics involved.

The core of a white dwarf is dense enough that electrons are subject to quantum mechanical effects, specifically the Pauli exclusion principle, that prevents the electrons from occupying the same quantum states [30]. The result is that electrons are forced into higher energy states, and the reaction to this forcing acts as a pressure that supports the star. Matter in this high-density condition is said to be "degenerate," and the case of interest, a white dwarf that has gained mass from a companion, is known as the "single degenerate" scenario.

If a white dwarf gains mass from a binary companion, the central density and temperature increase, and the star may eventually collapse to a neutron star. For a white dwarf composed principally of carbon and oxygen, before the collapse can occur the density and temperature reach values at which carbon fusion begins and the star enters a period of simmering that drives convection in the core producing a growing convective zone [31,32]. After $\approx 10^3$ yr the local temperature is hot enough that the burning timescale becomes shorter than a convective turnover time so that a local patch runs away and a subsonic flame is born, initiating the explosion [32].

Early one-dimensional simulations of the single-degenerate case showed that the most successful scenario follows the initial deflagration (subsonic reaction front) with a (supersonic) detonation, i.e. a deflagration-detonation transition (DDT [33,34]). Models with such a delayed detonation naturally account for some spectral features and the chemical stratification observed in the ejecta [35]. While one-dimensional models are able to reproduce observed features of the light curve and spectra, much of the physics is missing. Of particular concern is the degree to which the white dwarf expands during the deflagration phase of the explosion, which multidimensional simulations show depends on the behavior of fluid instabilities at the flame front. The degree of expansion during the deflagration phase is critical to the explosion because it determines the density at which the majority of the stellar material burns, which in turn controls the nucleosynthetic yield. Capturing the effects of fluid instabilities is therefore essential to modeling this process and necessitates the development of multidimensional models. By relaxing the symmetry constraints on the model, buoyancy instabilities are naturally captured leading to a strong dependence on the initial conditions of the deflagration. Some cases with a DDT criterion based on previous one-dimensional studies indicated the result is too little expansion of the star prior to the detonation [36,37,38,39]. Multidimensional models, however, may reach the expected amount of expansion prior to the DDT with the choice of particular ignition conditions and thus retain the desirable features from one-dimensional models [40,41,42,43]. Our investigation centers on how properties of the progenitor white dwarf that follow from properties of the host galaxy or its evolutionary history influence this DDT scenario and hence the brightness of an event measured by the ^{56}Ni yield.

2. The Systematics of SNe Ia in the DDT Scenario

In Townsley et al. (2009) [44] we investigated the direct effect of reprocessed stellar material (metals) in the host galaxy via the initial neutron excess of the progenitor white dwarf. Because of weak interactions, metals produced by nuclear burning are more neutron rich than hydrogen and helium, and, accordingly, the neutron excess of these elements is thought to drive the explosion yield toward stable iron-group elements. Thus, there is relatively less radioactive ^{56}Ni in the NSE mix, which results in a dimmer event [45]. We investigated this effect by introducing ^{22}Ne into the progenitor white dwarf as a proxy for (neutron-rich) metals. The presence of ^{22}Ne influences the progenitor structure, the energy release of the burn, and the flame speed. The study was designed to measure how the ^{22}Ne content influences the competition between rising plumes and the expansion of the star, which determines the yield. We performed a suite of 20 DDT simulations varying only the initial ^{22}Ne in a progenitor model, and found a negligible effect on the pre-detonation expansion of the star and thus the yield of NSE material. The neutron excess sets the amount of material in NSE that favors stable

iron-group elements over radioactive ^{56}Ni. Our results were consistent with earlier work calculating the direct modification of ^{56}Ni mass from initial neutron excess [45].

In Jackson et al. (2010) [46] we expanded the Townsley et al. study to include the indirect effect of metallicity in the form of the ^{22}Ne mass fraction through its influence on the density at which the DDT takes place. We simulated 30 realizations each at 5 transition densities between $1 - 3 \times 10^7$ g cm^{-3} for a total of 150 simulations. We found a quadratic dependence of the NSE yield on the log of the transition density, which is determined by the competition between rising unstable plumes and stellar expansion. By then considering the effect of metallicity on the transition density, we found the NSE yield decreases slightly with metallicity, but that the ratio of the 56Ni yield to the overall NSE yield does not change as significantly. Observations testing the dependence of the yield on metallicity remain somewhat ambiguous, but the dependence we found is comparable to that inferred from [47]. We also found that the scatter in the results increases with decreasing transition density, and we attribute this increase in scatter to the nonlinear behavior of the unstable rising plumes.

In Krueger et al. (2010) [48] we investigated the effect of central density on the explosion yield. We found that the overall production of NSE material did not change, but there was a definite trend of decreasing ^{56}Ni production with increasing progenitor central density. We attribute this result to higher rates of weak interactions (electron captures) that produce a higher proportion of neutronized material at higher density. More neutronization means less symmetric nuclei like ^{56}Ni, and, accordingly, a dimmer event. This result may explain the observed decrease in SNe Ia brightness with increasing delay time. The central density of the progenitor is determined by its evolution, including the transfer of mass from the companion. If there is a long period of cooling before the onset of mass transfer, the central density of the progenitor will be higher when the core reaches the carbon ignition temperature, thereby producing less ^{56}Ni and thus a dimmer event. In addition, we found considerable variation in the trends from some realizations, stressing the importance of statistical studies.

Our approach in these investigations has been to isolate facets of the problem expected to follow from properties of the host galaxy or evolution history of the progenitor white dwarf and perform a controlled statistical analysis of a ensemble of multidimensional simulations (described below). Not surprisingly, many other possible systematic effects exist (outlined in [44]) that were held fixed in each study. The ultimate goal of our research into thermonuclear supernovae is to consider the interdependence of all of these effects in the construction of the full theoretical picture.

3. Methodology

Our methodology for the theoretical study of thermonuclear supernovae consists of four principal parts. First is the ability to construct parameterized, hydrostatic initial white dwarf progenitors that can freely change thermal and compositional structure to match features from the literature about progenitor models [46]. Second is a model flame and energetics scheme with which to track both (subsonic) deflagrations and (supersonic) detonations as well as the evolution of dynamic ash in NSE. This flame/energetics scheme is implemented in the Flash hydrodynamics code [49,50,51]. Third is utilization of a scheme to post-process the density and temperature histories of Lagrangian tracer particles with a detailed nuclear network in order to calculate detailed nucleosynthetic yields [52,53]. Fourth, we developed a statistical framework with which to perform ensembles of simulations for well-controlled studies of systematic effects. Below we highlight the flame model and the statistical framework. Complete details of the methodology can be found in previously published results [52,54,55,44,53].

3.1. Flame Model

The great disparity between the length scale of a white dwarf (~10^9 cm) and the width of laminar nuclear flame (<1 cm) necessitates the use of a model flame in simulations of thermonuclear supernovae. Even simulations with adaptive mesh refinement cannot resolve the actual diffusive flame front in a simulation of the event. The model we use propagates an artificially broadened flame front with an advection-diffusion-reaction (ADR) scheme [56,57] that has been demonstrated to be

acoustically quiet and produce a unique flame speed [55]. This scheme evolves a reaction progress variable ϕ, where $\phi = 0$ indicates unburned fuel and $\phi = 1$ indicates burned ash, with the advection-reaction-diffusion equation

$$\partial_t \phi + \vec{u} \cdot \nabla \phi = \kappa \nabla^2 \phi + \frac{1}{\tau} R(\phi) . \tag{1}$$

Here κ is a constant and $R(\phi)$ a non-dimensional function, and both are tuned to propagate the reaction front at the physical speed of the real flame [58,59] and to be just wide enough to be resolved in our simulation. We use a modified version of the KPP reaction rate discussed by [57], in which $R \propto (\phi - \varepsilon)(1 - \phi + \varepsilon)$, where $\simeq 10^{-3}$.

In simulating the deflagration phase, the flame front separates expanded burned material (the hot ash) from denser unburned stellar material (cold fuel). The expansion and buoyancy of the burned material forces the interface upward into the denser fuel, and the configuration is susceptible to the Rayleigh-Taylor fluid instability [60,61]. It is necessary to enhance the flame speed in order to prevent turbulence generated by the Raleigh-Taylor instability from destroying the artificially broadened flame front. In the simulations discussed here, the enhancement is accomplished by the method suggested by Khokhlov [56] in which we prevent the flame front speed, s, from falling below a threshold that is scaled with the strength of the Raleigh-Taylor instability on the scale of the flame front ($s \propto \sqrt{g\ell}$, where g is the local gravity and and ℓ is the width of the artificial flame, which is a few times the grid resolution). This scaling of the flame speed prevents the Rayleigh-Taylor instability from effectively pulling the flame apart and also mimics what is a real enhancement of the burning area that is occurring due to structure in the flame surface on unresolved scales. It is expected that, for much of the white dwarf deflagration, the flame is "self-regulated," in which the small scale structure of the flame surface is always sufficient to keep up with the large-scale buoyancy-driven motions of the burned material. Thus the actual burning rate is determined by this action, which is resolved in our simulations.

As seen in Figure 1, [55] and by [62], these techniques demonstrate a suitable level of convergence for studies such as ours. This technique does makes it necessary to explicitly drop s to zero below a density of 10^7 g cm^{-3}, approximately where the real flame will be extinguished. This prescription captures some effects of the Rayleigh-Taylor instability and maintains the integrity of the thickened flame, but it does not completely describe the flame-turbulence interaction. Also, we neglect any enhancement from background turbulence from convection prior to the birth of the flame. Future work will include physically-motivated models for these effects [63].

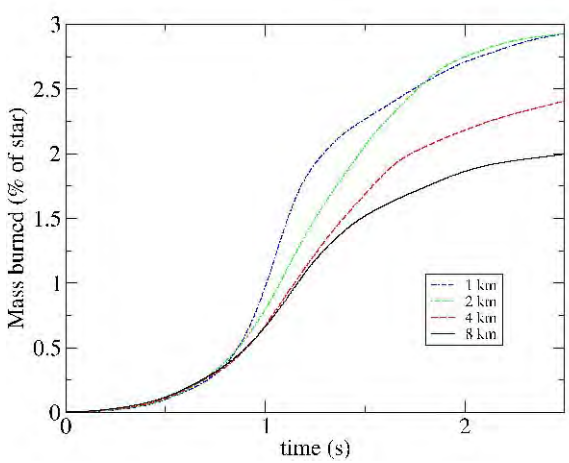


Figure 1. Fraction of star burned during the deflagration phase of a thermonuclear supernova for four two-dimensional simulations at varying resolutions. The initial conditions consisted of a 16 km ignition point started at a radius of 40 km from the center of the star. The result of this off-set ignition point is a single rising plume that may subsequently trigger a detonation [41]. Shown are effective resolutions of 8, 4, 2, and 1 km, and the results demonstrate reasonable convergence, especially during the early phases. Adapted from Townsley et al. (2009).

The model also includes the nuclear energy release occurring at the flame front and in the dynamic ash in NSE. We performed a detailed study of the nuclear processes occurring in a flame in the interior

of a white dwarf and developed an efficient and accurate method for incorporating the results into numerical simulations [54]. Tracking even tens of nuclear species is computationally prohibitive, and many more than this are required to accurately calculate the physics such as electron captures rates that are essential to studying rates of neutronization. We instead reproduce the energy release of the nuclear reactions with a highly abstracted model based on tabulation of properties of the burned material calculated in our study of the relevant nuclear processes.

The nuclear processing can be well approximated as a three stage process: initially carbon is consumed, followed by oxygen, which creates a mixture of silicon-group and light elements that is in nuclear statistical quasi-equilibrium (NSQE), finally the silicon-group nuclei are converted to iron-group, reaching full NSE. In both of these equilibrium states, the capture and creation of light elements (via photodisintegration) is balanced, so that energy release can continue by changing the relative abundance of light (low nuclear binding energy) and heavy (high nuclear binding energy) nuclides, an action that releases energy as buoyant burned material rises and expands. We track each of these stages with separate progress variables and separate relaxation times derived from full nuclear network calculations [54]. We define three progress variables representing consumption of carbon ϕ_C, consumption of oxygen to material in NSQE, ϕ_{NSQE}, and conversion of silicon to iron group nuclides to form true NSE material, ϕ_{NSE}. The physical state of the fluid is tracked with the electron number per baryon, Y_e, the number of nuclei per baryon, Y_i, and the average binding energy per baryon, $\bar{q}$, the minimum properties necessary to hydrodynamically evolve the fluid. Carbon consumption is coupled directly to the flame progress variable, $\phi_C \equiv \phi$, from eq. (1) above, and the later flame stages follow using simple relationships from more detailed calculations.

Burning and evolution of post-flame material change the nuclear binding energy, and we use the binding energy of magnesium to approximate the intermediate burning products of carbon. The method is finite differenced in such a way to ensure explicit conservation of energy. Weak processes (e.g., electron capture) are included in the calculation of the energy input rate, as are neutrino losses, which are calculated by convolving the NSE distribution with the weak interaction cross sections. Tracking the conversion of silicon-group nuclides to the iron-group is important for studying the effects of electron capture because the thresholds are lower for the iron-group nuclides. Both the NSE state and the electron capture rates were calculated with a set of 443 nuclides including all which have weak interaction cross sections given by [64]. This treatment of electron capture in the energy release is the most realistic currently in use for thermonuclear supernovae. Electron capture feeds back on the hydrodynamics in three ways: the NSE can shift to more tightly bound elements as the electron fraction, Y_e, changes, releasing some energy and changing the local temperature; also the reduction in Y_e lowers the Fermi energy, reducing the primary pressure support of this highly degenerate material and having an impact on the buoyancy of the neutronized material; finally neutrinos are emitted (since the star is transparent to them) so that some energy is lost from the system.

In addition to all of these effects during the deflagration phase of SNe Ia, the progress-variable-based method has been extended to model the gross features of detonations [65,53]. Instead of coupling the first burning stage, ϕ_C, representing carbon consumption, to the ADR flame front, we instead can use the actual temperature-dependent rate for carbon burning, or possibly a more appropriate effective rate. Doing so allows shock propagation to trigger burning and therefore create a propagating detonation. This method has been used successfully by [66] in modern studies of the DDT, and our multistage burning model shares many features with theirs (see also [67] and [68]). We treat the later burning stages very similarly, though we have taken slightly more care to track the intermediate stages and have nearly eliminated acoustic noise when coupling energy release to the flame.

Results from a two-dimensional DDT simulation utilizing these capabilities are shown in Figure 2. The three panels illustrate the early deflagration phase, the configuration just prior to ignition of the first detonation, and the progress of the detonation. The blue contour marks the division between the pre-explosion convective core, which has C/O/^{22}Ne composition of 30/66/4% by mass, and an outer

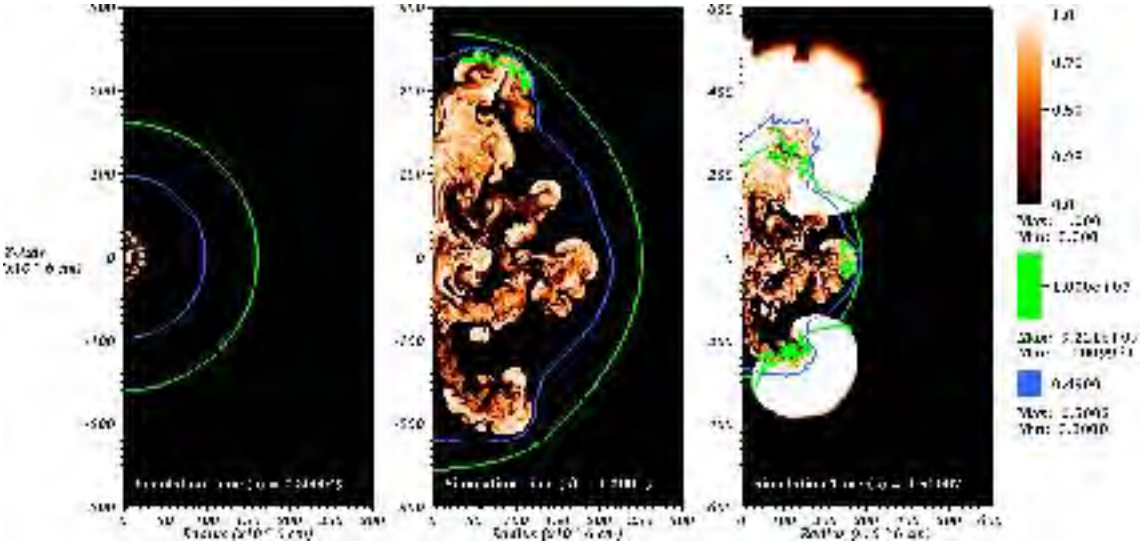


Figure 2. Images from a two-dimensional SNe Ia simulation from a neutronized-core progenitor. The left panel shows the development of fluid instabilities during the early deflagration phase, the center panel shows the configuration just prior to the first detonation, and the right panel shows the configuration with two distinct detonations consuming the star. Shown in color scale is the carbon-burning reaction progress variable which evolves from 0 to 1 and contours of $\rho = 10^7$ g cm^{-3} (green) and where the initial $X_{^{12}C} = 0.49$ (blue). The latter, initially inner, contour indicates the separation between the convective, low C-abundance, neutronized core and the higher C surface layers. Note that the scale on the right panel is twice that of the first two in order to accommodate the expansion of the star.

region with a composition of 50/48/2%. The contrast in C/O abundance reflects the stellar evolution of the progenitor, during which the central regions are produced by convective core helium burning, while the outer layers are produced by shell burning during the asymptotic giant branch phase [69]. The contrast in ^{22}Ne is due to production during the simmering phase.

3.2. Statistical Framework

We also developed a theoretical framework for the study of systematic effects in SNe Ia that will utilize two-and three-dimensional simulations in the DDT paradigm [44]. This framework allows the evaluation of the average dependence of the properties of SNe Ia on underlying parameters, such as composition, by constructing a theoretical sample based on a probabilistic initial ignition condition. Such sample-averaged dependencies are important for understanding how SN Ia models may explain features of the observed sample, particularly samples generated by large dark energy surveys utilizing SNe Ia as distance indicators.

The theoretical sample is constructed to represent statistical properties of the observed sample of SNe Ia such as the mean inferred ^{56}Ni yield and variance. Within the DDT paradigm, the variance in ^{56}Ni yields can be explained by the development of fluid instabilities during the deflagration phase of the explosion. By choice of the initial configuration of the flame, we may influence the growth of these fluid instabilities resulting in varying amounts of ^{56}Ni synthesized during the explosion. [44] found that perturbing a spherical flame surface with radius (r_0 = 150 km) with spherical harmonic modes (Y_l^m) between $12 \leq l \leq 16$ with random amplitudes (A) normally distributed between 0–15 km and, for 3D, random phases (δ) uniformly distributed between $-\pi - \pi$ best characterized the mean inferred ^{56}Ni yield and sample variance from observations:

$$r(\theta) = r_0 + \sum_{l=l_{\min}}^{l_{\max}} A_l Y_l(\theta) \tag{2}$$

$$r(\theta,\phi) = r_0 + \sum_{l=l_{\min}}^{l_{\max}} \sum_{m=-l}^{l} \frac{A_l^m e^{i\delta_l^m}}{\sqrt{2l+1}} Y_l^m(\theta,\phi). \tag{3}$$

With a suitable random-number generator, a sample population of progenitor WDs is constructed by defining the initial flame surface for a particular progenitor.

4. Statistical properties of the simulations

For this contribution, we focus on the ensemble of simulations from the central density study. In this study, the measure of the explosion is the yield of radioactive ^{56}Ni, the decay of which powers the light curve. In comparing a simulated explosion to astronomical observations, the results from simulations must be considered at frequencies of light corresponding to the bands in which the observations are made. We applied a simple relationship between the mass of ^{56}Ni synthesized in the explosion and the peak brightness in the V band. Thus the assumption for these models that the amount of ^{56}Ni synthesized in the explosion directly corresponds to the peak V-band brightness [24,70,71].

In the study, we found considerable variation in the amount of synthesized ^{56}Ni that we attribute to differences in the evolution during the deflagration phase of the explosion. As described above, the initial conditions for a realization were established stochastically. The deeply non-linear behavior of the subsequent evolution the buoyant rising plumes results in considerable variation in the duration of the deflagration phase, which is set by the time required for the first plume to rise to the DDT density. During the deflagration, the star reacts to the subsonic burning and expands. When a plume reaches the DDT threshold, the subsequent detonation very rapidly incinerates the expanded star (see Figure 2 for an illustration). ^{56}Ni is synthesized when stellar material burns at relatively high densities, so the amount of expansion determines the mass of material that will burn to ^{56}Ni. Thus the evolution during the deflagration, particularly its duration, strongly influences the ^{56}Ni yield. From this "noisy" background we were able to find a trend of decreased ^{56}Ni yield with increasing central density.

Figure 3 plots the mass of ^{56}Ni as a function of central density, illustrating the trend and scatter in our ensemble of simulations. Shown are the average among realizations, the standard deviation, and the standard error. Note that the standard deviation of our sample is about as large as the limits of the trend. Realizations 2, 4, and 8 in the figure exemplify the variation with progenitor central density, specifically the non-monotonicity of the individual trends. While these three realizations show a decreasing standard deviation with increasing central density, when considering the entire sample population, the standard deviation remains approximately the same as a function of central density. Table 1 provides the minima, mean, maxima, and standard deviations for ^{56}Ni and NSE masses at each central density.

In order to obtain a statistically meaningful trend, we must first demonstrate that our sample size characterizes the properties of the population. In Figure 4, we show that the standard deviation of the ^{56}Ni mass converges for all progenitor central densities with 15–20 realizations. Second, we must perform a number of realizations (take enough samples) such that the standard error of the mean is small enough to produce a statistically meaningful average trend. In the case that no trend exists, the number or realizations provides a limit to the magnitude of the mean trend. The standard error of the mean is computed to be the standard deviation divided by the square root of the number of samples.

We analyzed the trend of ^{56}Ni mass with progenitor central density with 30 realizations. While not shown graphically, the NSE mass does not vary significantly with central density. We evaluate an upper limit to the trend to be a decrease of 0.006 solar masses of NSE per 10^9 g cm^{-3} of central density. That result translates to less than a 0.6% change in the NSE yield per 10^9 g cm^{-3}. From Figures 3 and 4, we conclude that a statistical approach is necessary to evaluate systematic trends in highly nonlinear problems, such as SNe Ia.

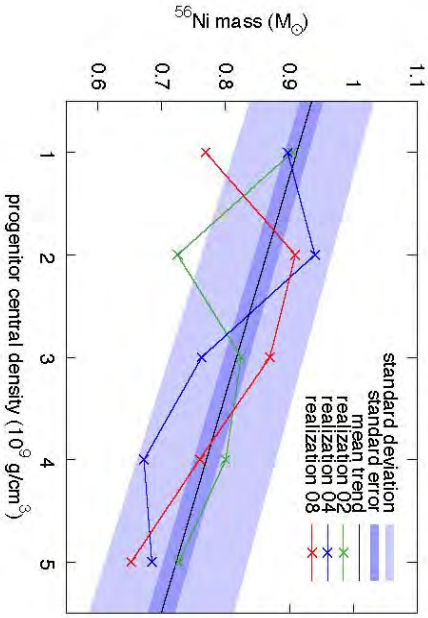


Figure 3. Relationship between mass of ^{56}Ni produced in a simulated event and central density of the progenitor white dwarf when the deflagration is ignited. The light purple is the standard deviation, the dark purple is the standard error of the mean, and the black line is the mean trend. Also shown are the trends for three different initial configurations as the central density varies. These three configurations do not match the statistical trend, nor are they monotonic; from this we conclude that a statistical study is important when considering a highly nonlinear problem such as a SN Ia. Adapted from Krueger et al. (2010) [48].

Table 1. Minima, means (μ), maxima, and standard deviations (σ) of ^{56}Ni and NSE masses from central density study. All masses are in units of M8.

ρ_c (g/cm^3)		1.000E+9	2.000E+9	3.000E+9	4.000E+9	5.000E+9
^{56}Ni	min	6.920E–1	6.905E–1	5.657E–1	5.828E–1	5.304E–1
	μ	9.105E–1	8.723E–1	8.035E–1	7.713E–1	7.255E–1
	max	1.052E+0	1.034E+0	9.813E–1	9.640E–1	9.288E–1
	σ	9.638E–2	9.945E–2	1.109E–1	1.091E–1	1.078E–1
NSE	min	7.969E–1	8.565E–1	7.783E–1	8.450E–1	8.197E–1
	μ	1.032E+0	1.050E+0	1.028E+0	1.038E+0	1.026E+0
	max	1.187E+0	1.225E+0	1.216E+0	1.228E+0	1.231E+0
	σ	1.039E–1	1.044E–1	1.149E–1	1.101E–1	1.082E–1

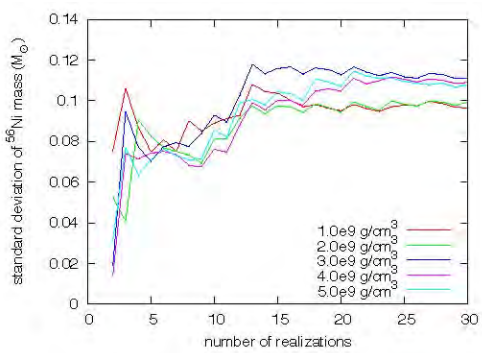


Figure 4. Convergence of the standard deviation of the mass as the data ^{56}Ni sample size increases for each of the five progenitor central densities. Each "realization" constitutes a unique initial condition configuration. Around 15–20 realizations achieve approximately the final standard deviation. The convergence of the standard deviation suggests that a sufficient number of data points have been included to compute a statistically correct mean.

5. Conclusions

Many studies have shown that small changes to the chosen initial conditions of a SNe Ia model result in large changes to the outcome of the explosion due to the non-linear effect of fluid instabilities. Our statistical framework relies on this property of multi-dimensional SNe Ia simulations to produce a sample population representative of the observed population. However, for the first time, we have

demonstrated that these small changes to the initial conditions can also influence systematic trends with properties of the progenitor WD such as central density. The effect of varying central density within a single realization does not characterize the entire sample population! In fact, each realization's outcome varies in a different way with changing central density as shown in Figure 3. It is only when all realizations in the sample population are considered that a meaningful and statistically significant trend emerges.

By varying the central density in a progenitor WD model, we also vary the density structure. The speed of the laminar flame depends on the density of the fuel it consumes. In hindsight, it is not that surprising that varying the fuel density of the initial flame results in non-linear behavior of the resulting supernovae similar to the effect of varying the initial position of the flame surface. Therefore, we must use a statistical ensemble of simulations to evaluate systematic trends in multi-dimensional SNe Ia that are subject to fluid instabilities.

In future studies we plan to explore systematic effects due to varying the core carbon-tooxygen (C/O) ratio. The core C/O ratio is thought to vary with zero-age main sequence mass of the progenitor WD, metallicity, and the mass of the companion star [72]. Because the laminar flame speed also depends on the carbon abundance in the fuel, a statistical ensemble of simulations will be necessary to evaluate this effect. It is not clear whether the results of [42] are characteristic of the observed sample of SNe Ia because a statistical approach was not employed.

Acknowledgments

This work was supported by the Department of Energy through grants DE-FG02-07ER41516, DE-FG02-08ER41570, and DE-FG02-08ER41565, and by NASA through grant NNX09AD19G. ACC also acknowledges support from the Department of Energy under grant DE-FG0287ER40317. DMT received support from the Bart J. Bok fellowship at the University of Arizona for part of this work. The authors acknowledge the hospitality of the Kavli Institute for Theoretical Physics, which is supported by the NSF under grant PHY05-51164, during the programs "Accretion and Explosion: the Astrophysics of Degenerate Stars" and "Stellar Death and Supernovae." The software used in this work was in part developed by the DOE-supported ASC/Alliances Center for Astrophysical Thermonuclear Flashes at the University of Chicago. This work was also supported in part by the US Department of Energy, Office of Nuclear Physics, under contract DE-AC02-06CH11357 and utilized resources at the New York Center for Computational Sciences at Stony Brook University/Brookhaven National Laboratory which is supported by the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 and by the State of New York.

References

[1] A. V. Filippenko. Optical spectra of supernovae. *Annu. Rev. Astron. Astrophys*, 35:309–355, 1997.

[2] W. Hillebrandt and J. C. Niemeyer. Type Ia supernova explosion models. *Annu. Rev. Astron. Astrophys*, 38:191, 2000.

[3] M. M. Phillips. The absolute magnitudes of type Ia supernovae. *Astrophysical Journal Letters*, 413:L105, 1993.

[4] A. G. Riess, W. H. Press, and R. P. Kirshner. A Precise Distance Indicator: Type IA Supernova Multicolor Light-Curve Shapes. *Astrophysical Journal*, 473:88–+, December 1996.

[5] A. Albrecht, G. Bernstein, R. Cahn, W. L. Freedman, J. Hewitt, W. Hu, J. Huth, M. Kamionkowski, E. W. Kolb, L. Knox, J. C. Mather, S. Staggs, and N. B. Suntzeff. Report of the Dark Energy Task Force. *ArXiv Astrophysics e-prints*, astro-ph/0609591, September 2006.

[6] A. G. Riess, A. V. Filippenko, P. Challis, A. Clocchiatti, A. Diercks, P. M. Garnavich, R. L. Gilliland, C. J. Hogan, S. Jha, R. P. Kirshner, B. Leibundgut, M. M. Phillips, D. Reiss, B. P. Schmidt, R. A. Schommer, R. C. Smith, J. Spyromilio, C. Stubbs, N. B. Suntzeff, and J. Tonry. Observational evidence from supernovae for an accelerating universe and a cosmological constant. *Astronomical Journal*, 116:1009, 1998.

[7] S. Perlmutter, G. Aldering, G. Goldhaber, R. A. Knop, P. Nugent, P. G. Castro, S. Deustua, S. Fabbro, A. Goobar, D. E. Groom, I. M. Hook, A. G. Kim, M. Y. Kim, J. C. Lee, N. J. Nunes, R. Pain, C. R. Pennypacker, R. Quimby, C. Lidman, R. S. Ellis, M. Irwin, R. G. McMahon, P. Ruiz-Lapuente, N. Walton, B. Schaefer, B. J. Boyle, A. V. Filippenko, T. Matheson, A. S. Fruchter, N. Panagia, H. J. M. Newberg, W. J. Couch, and The Supernova Cosmology Project. Measurements of Omega and Lambda from 42 high-redshift supernovae. *Astrophysical Journal*, 517:565, 1999.

[8] E. Kolb et al. Report of the Dark Energy Task Force. electronic, June 2006. http://www.science.doe.gov/hep/DETF-FinalRptJune30,2006.pdf.

[9] M. Hicken, W. M. Wood-Vasey, S. Blondin, P. Challis, S. Jha, P. L. Kelly, A. Rest, and R. P. Kirshner. Improved Dark Energy Constraints from ˜100 New CfA Supernova Type Ia Light Curves. *Astrophysical Journal*, 700:1097–1140, August 2009.

[10] H. Lampeitl, R. C. Nichol, H.-J. Seo, T. Giannantonio, C. Shapiro, B. Bassett, W. J. Percival, T. M. Davis, B. Dilday, J. Frieman, P. Garnavich, M. Sako, M. Smith, J. Sollerman, A. C. Becker, D. Cinabro, A. V. Filippenko, R. J. Foley, C. J. Hogan, J. A. Holtzman, S. W. Jha, K. Konishi, J. Marriner, M. W. Richmond, A. G. Riess, D. P. Schneider, M. Stritzinger, K. J. van der Heyden, J. T. Vanderplas, J. C. Wheeler, and C. Zheng. First-year Sloan Digital Sky Survey-II supernova results: consistency and constraints with other intermediate-redshift data sets. *Monthly Notices of the Royal Astronomical Society*, 401:2331–2342, February 2010.

[11] R. P. Kirshner. Foundations of Supernova Cosmology. *ArXiv e-prints*, October 2009.

[12] J. W. Truran and A. G. W. Cameron. Evolutionary Models of Nucleosynthesis in the Galaxy. *Ap&SS*, 14:179–222, November 1971.

[13] D. A. Howell, A. Conley, M. Della Valle, P. E. Nugent, S. Perlmutter, G. H. Marion, K. Krisciunas, C. Badenes, P. Mazzali, G. Aldering, P. Antilogus, E. Baron, A. Becker, C. Baltay, S. Benetti, S. Blondin, D. Branch, E. F. Brown, S. Deustua, A. Ealet, R. S. Ellis, D. Fouchez, W. Freedman, A. Gal-Yam, S. Jha, D. Kasen, R. Kessler, A. G. Kim, D. C. Leonard, W. Li, M. Livio, D. Maoz, F. Mannucci, T. Matheson, J. D. Neill, K. Nomoto, N. Panagia, K. Perrett, M. Phillips, D. Poznanski, R. Quimby, A. Rest, A. Riess, M. Sako, A. M. Soderberg, L. Strolger, R. Thomas, M. Turatto, S. van Dyk, and W. M. Wood-Vasey. Type Ia supernova science 2010–2020. *ArXiv e-prints*, March 2009. 0903.1086.

[14] N. Gehrels. Report on the science coordination group activities for the joint dark energy mission (jdem). Technical report, Goddard Space Flight Center, 2009.

[15] R. A. Scalzo, G. Aldering, P. Antilogus, C. Aragon, S. Bailey, C. Baltay, S. Bongard, C. Buton, M. Childress, N. Chotard, Y. Copin, H. K. Fakhouri, A. Gal-Yam, E. Gangler, S. Hoyer, M. Kasliwal, S. Loken, P. Nugent, R. Pain, E. Pécontal, R. Pereira, S. Perlmutter, D. Rabinowitz, A. Rau, G. Rigaudier, K. Runge, G. Smadja, C. Tao, R. C. Thomas, B. Weaver, and C. Wu. Nearby Supernova Factory Observations of SN 2007if: First Total Mass Measurement of a Super-Chandrasekhar-Mass Progenitor. *Astrophysical Journal*, 713:1073–1094, April 2010.

[16] F. Yuan, R. M. Quimby, J. C. Wheeler, J. Vinkó, E. Chatzopoulos, C. W. Akerlof, S. Kulkarni, J. M. Miller, T. A. McKay, and F. Aharonian. The Exceptionally Luminous Type Ia Supernova 2007if. *Astrophysical Journal*, 715:1338–1343, June 2010.

[17] M. Gilfanov and Á. Bogdán. An upper limit on the contribution of accreting white dwarfs to the typeIa supernova rate. *Nature*, 463:924–925, February 2010.

[18] W. D. Arnett. Type I supernovae. I—analytic solutions for the early part of the light curve. *Astrophysical Journal*, 253:785, 1982.

[19] P. A. Pinto and R. G. Eastman. The physics of type Ia supernova light curves. I. analytic results and time dependence. *Astrophysical Journal*, 530:744–756, 2000.

[20] D. Kasen and S. E. Woosley. On the Origin of the Type Ia Supernova Width-Luminosity Relation. *Astrophysical Journal*, 656:661–665, February 2007.

[21] J. S. Gallagher, P. M. Garnavich, P. Berlind, P. Challis, S. Jha, and R. P. Kirshner. Chemistry and Star Formation in the Host Galaxies of Type Ia Supernovae. *Astrophysical Journal*, 634:210–226, November 2005.

[22] J. S. Gallagher, P. M. Garnavich, N. Caldwell, R. P. Kirshner, S. W. Jha, W. Li, M. Ganeshalingam, and A. V. Filippenko. Supernovae in Early-Type Galaxies: Directly Connecting Age and Metallicity with Type Ia Luminosity. *Astrophysical Journal*, 685:752–766, October 2008.

[23] J. D. Neill, M. Sullivan, D. A. Howell, A. Conley, M. Seibert, D. C. Martin, T. A. Barlow, K. Foster, P. G. Friedman, P. Morrissey, S. G. Neff, D. Schiminovich, T. K. Wyder, L. Bianchi, J. Donas, T. M. Heckman, Y.-W. Lee, B. F. Madore, B. Milliard, R. M. Rich, and A. S. Szalay. The Local Hosts of Type Ia Supernovae. *Astrophysical Journal*, 707:1449–1465, December 2009.

[24] D. A. Howell, M. Sullivan, E. F. Brown, A. Conley, D. LeBorgne, E. Y. Hsiao, P. Astier, D. Balam, C. Balland, S. Basa, R. G. Carlberg, D. Fouchez, J. Guy, D. Hardin, I. M. Hook, R. Pain, K. Perrett, C. J. Pritchet, N. Regnault, S. Baumont, J. LeDu, C. Lidman, S. Perlmutter, N. Suzuki, E. S. Walker, and J. C. Wheeler. The Effect of Progenitor Age and Metallicity on Luminosity and ^{56}Ni Yield in Type Ia Supernovae. *Astrophysical Journal*, 691:661–671, January 2009.

[25] B. K. Krueger, A. P. Jackson, D. M. Townsley, A. C. Calder, E. F. Brown, and F. X. Timmes. On Variations of the Brightness of Type Ia Supernovae with the Age of the Host Stellar Population. *Astrophysical Journal Letters*, submitted, 2010.

[26] F. Mannucci, M. Della Valle, and N. Panagia. Two populations of progenitors for Type Ia supernovae? *Monthly Notices of the Royal Astronomical Society*, 370:773–783, August 2006.

[27] C. Raskin, E. Scannapieco, J. Rhoads, and M. Della Valle. Prompt Ia Supernovae are Significantly Delayed. *Astrophysical Journal*, 707:74–78, December 2009.

[28] J. S. Gallagher, P. M. Garnavich, N. Caldwell, R. P. Kirshner, S. W. Jha, W. Li, M. Ganeshalingam, and A. V. Filippenko. Supernovae in Early-Type Galaxies: Directly Connecting Age and Metallicity with Type Ia Luminosity. *Astrophysical Journal*, 685:752–766, October 2008.

[29] T. D. Brandt, R. Tojeiro, É. Aubourg, A. Heavens, R. Jimenez, and M. A. Strauss. The Ages of Type Ia Supernova Progenitors. *ArXiv e-prints*, February 2010.

[30] R. H. Dicke and J. P. Wittke. *Introduction to Quantum Mechanics*. Addison-Wesley, Reading, 1960.

[31] S. E. Woosley, S. Wunsch, and M. Kuhlen. Carbon Ignition in Type Ia Supernovae: An Analytic Model. *Astrophysical Journal*, 607:921–930, June 2004.

[32] M. Zingale, A. S. Almgren, J. B. Bell, A. Nonaka, and S. E. Woosley. Low Mach Number Modeling of Type IA Supernovae. IV. White Dwarf Convection. *Astrophysical Journal*, 704:196–210, October 2009.

[33] A. M. Khokhlov. Delayed detonation model for type IA supernovae. *Astronomy and Astrophysics*, 245:114–128, May 1991.

[34] P. Höflich and A. Khokhlov. Explosion Models for Type IA Supernovae: A Comparison with Observed Light Curves, Distances, H 0, and Q 0. *Astrophysical Journal*, 457:500–+, February 1996.

[35] D. Kasen and T. Plewa. Spectral Signatures of Gravitationally Confined Thermonuclear Supernova Explosions. *Astrophysical Journal Letters*, 622:L41–L44, March 2005.

[36] J. C. Niemeyer, W. Hillebrandt, and S. E. Woosley. Off-Center Deflagrations in Chandrasekhar Mass Type IA Supernova Models. *Astrophysical Journal*, 471:903–+, November 1996.

[37] A. C. Calder, T. Plewa, N. Vladimirova, E. F. Brown, D. Q. Lamb, K. Robinson, and J. W. Truran. Deflagrating white dwarfs: a Type Ia supernova model. In *Bulletin of the American Astronomical Society*, volume 35 of *Bulletin of the American Astronomical Society*, pages 1278–+, December 2003.

[38] A. C. Calder, T. Plewa, N. Vladimirova, D. Q. Lamb, and J. W. Truran. Type Ia Supernovae: An Asymmetric Deflagration Model. *ArXiv Astrophysics e-prints*, May 2004. astro-ph/0405162.
[39] E. Livne, S. M. Asida, and P. Höflich. On the Sensitivity of Deflagrations in a Chandrasekhar Mass White Dwarf to Initial Conditions. *Astrophysical Journal*, 632:443–449, October 2005.
[40] V. N. Gamezo, A. M. Khokhlov, and E. S. Oran. Deflagrations and detonations in thermonuclear supernovae. *Physical Review Letters*, 92:211102, 2004.
[41] T. Plewa, A. C. Calder, and D. Q. Lamb. Type Ia supernova explosion: Gravitationally confined detonation. *Astrophysical Journal Letters*, 612:L37–L40, 2004.
[42] F. K. Röpke, M. Gieseler, M. Reinecke, C. Travaglio, and W. Hillebrandt. Type Ia Supernova Diversity in Three-Dimensional Models. *Astronomy and Astrophysics*, 453:203, 2006.
[43] G. C. Jordan, IV, R. T. Fisher, D. M. Townsley, A. C. Calder, C. Graziani, S. Asida, D. Q. Lamb, and J. W. Truran. Three-Dimensional Simulations of the Deflagration Phase of the Gravitationally Confined Detonation Model of Type Ia Supernovae. *Astrophysical Journal*, 681:1448–1457, July 2008.
[44] D. M. Townsley, A. P. Jackson, A. C. Calder, D. A. Chamulak, E. F. Brown, and F. X. Timmes. Evaluating Systematic Dependencies of Type Ia Supernovae: The Influence of Progenitor ^{22}Ne Content on Dynamics. *Astrophysical Journal*, 701:1582–1604, August 2009.
[45] F. X. Timmes, E. F. Brown, and J. W. Truran. On variations in the peak luminosity of type Ia supernovae. *Astrophysical Journal*, 590:L83, 2003.
[46] A. P. Jackson, A. C. Calder, D. M. Townsley, D. A. Chamulak, E. F. Brown, and F. X. Timmes. On variations of the brightness of type Ia supernovae with the age of the host stellar population. *Astrophysical Journal*, 2010. submitted.
[47] E. Bravo, I. Domínguez, C. Badenes, L. Piersanti, and O. Straniero. Metallicity as a Source of Dispersion in the SNIa Bolometric Light Curve Luminosity-Width Relationship. *Astrophysical Journal Letters*, 711:L66–L70, March 2010.
[48] B. K. Krueger, A. P. Jackson, D. M. Townsley, A. C. Calder, E. F. Brown, and F. X. Timmes. Evaluating Systematic Dependencies of Type Ia Supernovae: The Influence of Deflagration to Detonation Density. *Astrophysical Journal*, 2010. submitted.
[49] B. Fryxell, K. Olson, P. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, and H. Tufo. FLASH: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *Astrophysical Journal Supplement*, 131:273–334, 2000.
[50] A. C. Calder, B. C. Curtis, L. J. Dursi, B. Fryxell, G. Henry, P. MacNeice, K. Olson, P. Ricker, R. Rosner, F. X. Timmes, H. M. Tufo, J. W. Truran, and M. Zingale. High-performance reactive fluid flow simulations using adaptive mesh refinement on thousands of processors. In *Proceedings of Supercomputing 2000*, page http://sc2000.org, 2000.
[51] A. C. Calder, B. Fryxell, T. Plewa, R. Rosner, L. J. Dursi, V. G. Weirs, T. Dupont, H. F. Robey, J. O. Kane, B. A. Remington, R. P. Drake, G. Dimonte, M. Zingale, F. X. Timmes, K. Olson, P. Ricker, P. MacNeice, and H. M. Tufo. On validating an astrophysical simulation code. *Astrophysical Journal Supplement*, 143:201–229, 2002.
[52] E. F. Brown, A. C. Calder, T. Plewa, P. M. Ricker, K. Robinson, and J. B. Gallagher. Type Ia supernovae: Simulations and nucleosynthesis. *Nucl. Phys. A*, 758:451, 2005.
[53] D. M. Townsley, F. X. Timmes, A. P. Jackson, A. C. Calder, and Brown. Nucleosynthesis During a Deflagration Detonation Transition Supernova Type Ia. *Astrophysical Journal*, 2010. to be submitted.
[54] A. C. Calder, D. M. Townsley, I. R. Seitenzahl, F. Peng, O. E. B. Messer, N. Vladimirova, E. F. Brown, J. W. Truran, and D. Q. Lamb. Capturing the Fire: Flame Energetics and Neutronization for Type Ia Supernova Simulations. *Astrophysical Journal*, 656:313–332, February 2007.

[55] D. M. Townsley, A. C. Calder, S. M. Asida, I. R. Seitenzahl, F. Peng, N. Vladimirova, D. Q. Lamb, and J. W. Truran. Flame Evolution During Type Ia Supernovae and the Deflagration Phase in the Gravitationally Confined Detonation Scenario. *Astrophysical Journal*, 668:1118–1131, October 2007.
[56] A. M. Khokhlov. Propagation of turbulent flames in supernovae. *Astrophysical Journal*, 449:695, 1995.
[57] N. Vladimirova, G. Weirs, and L. Ryzhik. Flame capturing with an advection-reaction-diffusion model. *Combust. Theory Modelling*, 10(5):727–747, 2006.
[58] F. X. Timmes and S. E. Woosley. The conductive propagation of nuclear flames. I—Degenerate C + O and O + Ne + Mg white dwarfs. *Astrophysical Journal*, 396:649, 1992.
[59] D. A. Chamulak, E. F. Brown, and F. X. Timmes. The Laminar Flame Speedup by ^{22}Ne Enrichment in White Dwarf Supernovae. *Astrophysical Journal Letters*, 655:L93, February 2007.
[60] G. Taylor. The Instability of Liquid Surfaces when Accelerated in a Direction Perpendicular to their Planes. I. *Royal Society of London Proceedings Series A*, 201:192–196, March 1950.
[61] S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Dover, New York, 1981.
[62] V. N. Gamezo, A. M. Khokhlov, E. S. Oran, A. Y. Chtchelkanova, and R. O. Rosenberg. Thermonuclear supernovae: Simulations of the deflagration stage and their implications. *Science*, 299:77, 2003.
[63] A. P. Jackson, D. M. Townsley, and A. C. Calder. in preparation, 2010.
[64] K. Langanke and G. Mart´ınez-Pinedo. Rate tables for the weak processes of pf-shell nuclei in stellar environments. *At. Data Nucl. Data Tables*, 79:1, 2001.
[65] C. A. Meakin, I. Seitenzahl, D. Townsley, G. C. Jordan, J. Truran, and D. Lamb. Study of the Detonation Phase in the Gravitationally Confined Detonation Model of Type Ia Supernovae. *Astrophysical Journal*, 693:1188–1208, March 2009.
[66] V. N. Gamezo, A. M. Khokhlov, and E. S. Oran. Three-dimensional delayed-detonation model of type Ia supernovae. *Astrophysical Journal*, 623:337–346, April 2005.
[67] A. M. Khokhlov. Delayed detonation model for type Ia supernovae. *Astronomy and Astrophysics*, 245:114–128, 1991.
[68] A. M. Khokhlov. Three-Dimensional Modeling of the Deflagration Stage of a Type Ia Supernova Explosion. *Astrophysical Journal*, submitted, astro-ph/0008463, 2000.
[69] O. Straniero, I. Dom´ınguez, G. Imbriani, and L. Piersanti. The Chemical Composition of White Dwarfs as a Test of Convective Efficiency during Core Helium Burning. *Astrophysical Journal*, 583:878–884, February 2003.
[70] G. Goldhaber, D. E. Groom, A. Kim, G. Aldering, P. Astier, A. Conley, S. E. Deustua, R. Ellis, S. Fabbro, A. S. Fruchter, A. Goobar, I. Hook, M. Irwin, M. Kim, R. A. Knop, C. Lidman, R. McMahon, P. E. Nugent, R. Pain, N. Panagia, C. R. Pennypacker, S. Perlmutter, P. Ruiz-Lapuente, B. Schaefer, N. A. Walton, and T. York. Timescale Stretch Parameterization of Type Ia Supernova B-Band Light Curves. *Astrophysical Journal*, 558:359–368, September 2001.
[71] M. M. Phillips, P. Lira, N. B. Suntzeff, R. A. Schommer, M. Hamuy, and J. Maza. The reddening-free decline rate versus luminosity relationship for type Ia supernovae. *Astronomical Journal*, 118:1766–1776, 1999.
[72] H. Umeda, K. Nomoto, H. Yamaoka, and S. Wanajo. Evolution of 3–9 $M_{\odot}$ stars for $Z = 0.001$–0.03 and metallicity effects on type Ia supernovae. *Astrophysical Journal*, 513:861–868, 1999.

Towards exascale supernova simulations with GenASiS

Christian Y. Cardall,[1] Eirik Endeve,[1] Reuben D. Budiardja,[2] Pedro Marronetti,[3] and Anthony Mezzacappa[1]

[1]Physics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6354, USA
[2]Department of Physics and Astronomy, University of Tennessee, Knoxville, TN 37996-1200, USA
[3]Physics Department, Florida Atlantic University, Boca Raton, FL 33431-0991

Abstract. Core-collapse supernovae are worth simulating on leadership computing resources because they are the origin of many elements, are interesting targets for expensive observational programs, and provide valuable computational experience. The need to perform implicit neutrino transport addressing up to six dimensions of phase space makes simulations aimed at the explosion mechanism an exascale problem. The multiscale nature of the problem in space and time can be addressed with adaptive mesh refinement and implicit evolution. Several types of solvers—explicit and implicit, hyperbolic and possibly elliptic—must be deployed with operator splitting to address the multiphysics nature of the problem. While we have some ideas about how parallelism can be exploited, we look to applied mathematicians and computer scientists to work with us and provide libraries that deploy capabilities needed by our solvers on heterogeneous and communication-unfriendly computer architectures.

1. Why Simulate Supernovae?

Among all the applications of pressing practical importance facing the nation and the world, why should valuable leadership computing resources be spent on the simulation of supernovae? Practical questions are not the only ones that matter to us. Perhaps even more important—in the sense of having limited utility to anything else, and therefore having an intrinsic worth that answers to nothing 'higher'—are answers to questions relating to our origins and place in the universe.

How the building blocks of our bodies and our environment, the chemical elements, came to be formed and spread through the universe is one such question. As the saying goes, we are stardust: many important elements, or nuclear species, are synthesized by the nuclear fusion that powers stars. But it is not enough to synthesize the elements; these must also be dispersed if they are to be incorporated into planets and people. For stars with masses between $\sim 0.8\ M_\odot$ (where $M_\odot$ is the mass of the Sun) and $\sim 8\ M_\odot$, this dispersal is with a whimper as they puff their chemically-enriched envelopes into the interstellar medium. This can result in very beautiful structures, known historically, if semantically inaccurately, as planetary nebulae. In other cases the element dispersal can occur with a bang—one that includes further element formation (or 'nucleosynthesis'). Such an explosion is called a 'supernova,' which is shorthand for an 'exceptionally bright new star.' The peak optical luminosity of a supernova is comparable to that of its host galaxy containing billions of stars. SN 1987A is the most recent nearby supernova; it went off in the Large Magellanic Cloud, a small satellite galaxy to the Milky Way.

There are two basic physical scenarios that lead to supernovae. A thermonuclear supernova occurs when a white dwarf (remnant of a star with mass $M \lesssim 8\ M_\odot$) accretes too much mass from a binary

companion. Three thermonuclear supernovae have occurred in our Galaxy in historical times. A core-collapse supernova occurs when the core of a star with $M \gtrsim 8\ M_{\odot}$ collapses at the end of several burning stages, releasing gravitational energy. There have also been three core-collapse supernovae in the Milky Way in historical times. Modern surveys confirm what the small-number statistics of our own recent Galactic history suggest: thermonuclear and core-collapse supernovae occur with comparable frequency.

On a slightly more practical level, but still within the realm of purely scientific interest, core-collapse supernovae are interesting targets for observational programs in which large public investments have been made. These include telescopes—both earthbound and spaceborne—that cover various ranges of the electromagnetic spectrum. Astronomical observations investigate such aspects as the launch of an explosion; neutron star mass, spin, magnetic field, and kick velocity; composition of the ejecta; and explosion morphology. Even more exciting would be the detection of nontraditional astronomical signals from a supernova in our galaxy. Neutrinos are ghostly particles that interact only very weakly with ordinary matter; they are emitted in a certain type of nuclear decay, and from hot and dense nuclear matter, including in copious amounts from core-collapse supernovae. Gravitational waves are tiny ripples in spacetime radiated from violent events in which high-density matter moves at close to the speed of light. Both of these types of signals are so weak that it takes truly heroic experimental efforts to detect them. But that same weakness of interaction means that they escape their sources comparatively easily, so that, unlike photons, they bring us observational information directly from the heart of the supernova.

On an even more practical level, accomplishing core-collapse supernova simulations enlarges community experience in addressing several computational challenges of interest in other applied contexts. These include magnetohydrodynamics; reaction kinetics; particle transport; and treating all these in a multidimensional, multiscale, and multiphysics system.

2. Why is the Core-Collapse Supernova Mechanism an Exascale Problem?

Stars spend most of their lives burning hydrogen to helium in their cores. Stars with masses greater than $8\ M_{\odot}$ continue to burn up to oxygen, neon, and magnesium, and those heavier than $10\ M_{\odot}$ burn all the way up to iron group elements near the top of the nuclear binding energy curve. In the absence of core burning, electron degeneracy becomes the main source of pressure support. But once the electrons become relativistic they have nothing more to give. This is the basic physics behind the Chandrasekhar limit, and when the mass of the core exceeds it, dynamical collapse ensues.

The collapsing interior divides into an inner core, in sonic contact with itself, and an outer core whose infall is supersonic. Collapse of the inner core halts when the nuclear equation of state stiffens above nuclear density. But the outer core has not heard the news that collapse has halted. A shock wave forms when supersonically infalling material slams into the inner core. The shock moves out, heating the material through which it passes, and eventually will give rise to the optical emission we know as a supernova once it travels thousands of kilometers to the optically thin outermost stellar layers.

This does not happen right away, however. At around 150 or 200 km the shock stalls, due to a loss of pressure support, and becomes a ‘stationary accretion shock.’ This happens because of two enervating consequences of the high temperature of the shock-heated material: iron-group nuclei that fall through the shock are endothermically reduced to their constituent nucleons, and electron capture results in neutrino emission. The mechanism of shock revival—that is to say, the explosion mechanism—remains a subject of active investigation. But since the 1980s the delayed neutrino-driven explosion mechanism has been a primary paradigm: beyond the so-called ‘gain radius,’ heating by neutrino absorption outweighs cooling by electron capture, and on longer timescales (hundreds of milliseconds) may re-energize the shock [1].

Determining the heating and cooling rates that affect the fate of the shock requires neutrino transport. Deep inside the newly-born neutron star, where neutrinos are trapped and slowly diffuse outwards, their distribution is nearly isotropic. As they begin to decouple in the semitransparent

regime between the proto neutron star and the shock their angular distribution becomes more and more strongly forward-peaked. This transition happens differently for different neutrino energies. Ultimately, therefore, knowledge of the neutrino heating and cooling rates relies on knowledge of the neutrino distribution functions: at every instant in time and at every point in space, we would like to know how many neutrinos there are with a given energy moving in a given direction. This requires implicit solution of the Boltzmann equation or something equivalent.

All three spatial dimensions are important in core-collapse supernovae. Convection and related phenomena may occur in the proto neutron star, and there is convection in the region between the gain radius and the shock. Rotation and magnetic fields may also come into play. Aside from convection, rotation, and magnetic fields, it has been discovered in recent years that the stationary accretion shock is unstable. This stationary accretion shock instability, or SASI, may be responsible for phenomena previously attributed to rapid rotation of the progenitor star, including aspherical explosion morphology, pulsar spin, and magnetic field generation [2,3,4].

The high dimensionality of phase space—which, again, in its full glory consists of three spatial dimensions plus three momentum space dimensions—is, together with the need to perform implicit neutrino transport, most of what makes this an exascale problem. (Evolving a nuclear reaction network with many dozens of species at every spatial cell would also be overwhelming.) As an example, consider a very simpleminded estimate of how easily the inversion of dense blocks arising from momentum space coupling can exhaust exascale resources. Each dense block has $N_{\mathbf{p}}$ rows and columns, where $N_{\mathbf{p}} = N_\nu N_E N_\vartheta N_\varphi$ is the number of momentum space bins arising from N_ν neutrino species, N_E energy bins, and $N_\vartheta N_\varphi$ angle bins. Suppose the structure of the dense blocks can be exploited in such a way that their inversion can be solved in order $N_{\mathbf{p}}^2$ operations rather than the $N_{\mathbf{p}}^3$ operations that would be required by LU decomposition. There is one dense block for each of N_x spatial cells. Suppose this overall solve requires N_{it} iterations for each of N_t time steps. Then the total number of required flops goes like

$$N_{\mathrm{FLOP}} \sim N_t\, N_{\mathrm{it}}\, N_{\mathbf{x}}\, N_{\mathbf{p}}{}^2.$$

The wall time is

$$T_{\mathrm{wall}} = \frac{N_{\mathrm{FLOP}}}{\epsilon_{\mathrm{FLOP}} R_{\mathrm{FLOP}}},$$

where the efficiency $\varepsilon_{\mathrm{FLOP}}$ is the achieved fraction of the theoretical maximum FLOP rate R_{FLOP}. Plugging possible numbers, we arrive at the figure of merit

$$T_{\mathrm{wall}} \sim 7 \text{ weeks} \left(\frac{N_t}{10^6}\right) \left(\frac{N_{\mathrm{it}}}{20}\right) \left(\frac{N_{\mathbf{x}}}{10^6}\right) \left(\frac{N_{\mathbf{p}}}{10^5}\right)^2 \left(\frac{R_{\mathrm{FLOP}}}{10^{18}\mathrm{s}^{-1}}\right)^{-1} \left(\frac{\epsilon_{\mathrm{FLOP}}}{0.05}\right)^{-1}.$$

We see that exascale resources can be exhausted for several weeks for a single simulation.

Management and analysis of supernova simulation output is expected represent its own set of challenges on exascale machines. Long time integrations of large-scale, high-resolution simulations describing physics on wide range of spatial and temporal scales result in massive amounts of data being written to disk at frequent time intervals for analysis and restart. (Our current suite of high-resolution MHD simulations has produced hundreds of terabytes of simulation data, and is already posing challenges for post-simulation storage and data mining.) The additional physics required for a credible supernova model (i.e., relativistic gravity, neutrino momentum space, and nuclear kinetics) is expected to dramatically increase simulation data output. Our traditional (linear) operating mode of simulation execution followed by data analysis will likely have to be revised. Much of the expensive interaction between memory and disk may be reduced by carrying out basic visualization and analysis while simulation data is in memory, but initial simulations will explore new territory in supernova theory, and significant amounts of data is expected to be written and retained for post processing as well.

Given these overwhelming requirements, ways have been found by several groups (see for instance [5,6,7,8,9,10] and references therein) to perform more tractable computations by reducing the dimensionality of phase space in various ways. Given the different possible ways of reducing the dimensionality of both space and momentum space, not to mention different physics implementations and numerical methods, there has not been complete convergence of simulation outcomes over time.

The Oak Ridge/Florida Atlantic/North Carolina State collaboration, with which we are associated, has two major projects. One of them, CHIMERA [5], has recently seen explosions in axisymmetry with energy-dependent neutrino transport. A corresponding three-dimensional full multiphysics run with this code has also been started. The other major project of our collaboration, GenASiS, is ultimately aimed at addressing the full dimensionality of the neutrino transport problem, with reasonable intermediate steps.

3. How Can the Multiscale Nature of the Problem be Addressed?

Several things make a core-collapse supernova a multiscale problem. Perhaps the first and most obvious is that gravitational collapse changes the length scale by roughly a factor of 100. There are also critical surfaces that should be well resolved: the large density gradient at the proto neutron star surface; the shock; and interfaces between regions of different compositions.

Then there is the matter of turbulent cascades to small scales, which can make a difference. Small-scale turbulence may increase the dwell time of fluid elements in the 'gain region,' increasing their heating by neutrino emission. Moreover, the outcome of simulated magnetic field generation depends strongly on resolution. In the case of the magneto-rotational instability this is because the linear growth rate of unstable modes is inversely proportional to their wavelength. Another amplification mechanism operates in simulations we have performed with GenASiS [4]: turbulence tangles and stretches magnetic flux tubes until their thickness is comparable to the spatial resolution. Figure 1 shows slices through two 3D MHD simulations that differ by a factor of two in resolution; the difference in the amount of yellow visually betrays the difference in magnetic field strength. The dependence of average magnetic field strength on resolution is illustrated more quantitatively in Figure 2, which shows the behavior expected of this simple geometric effect in 2D and 3D. These simulations are still unresolved even at 1024^3 on over 32,768 cores. (As an aside, we note that GenASiS magnetohydrodynamics scaling has now been pushed to 1280^3 on 64,000 cores, as illustrated in Figure 3.)

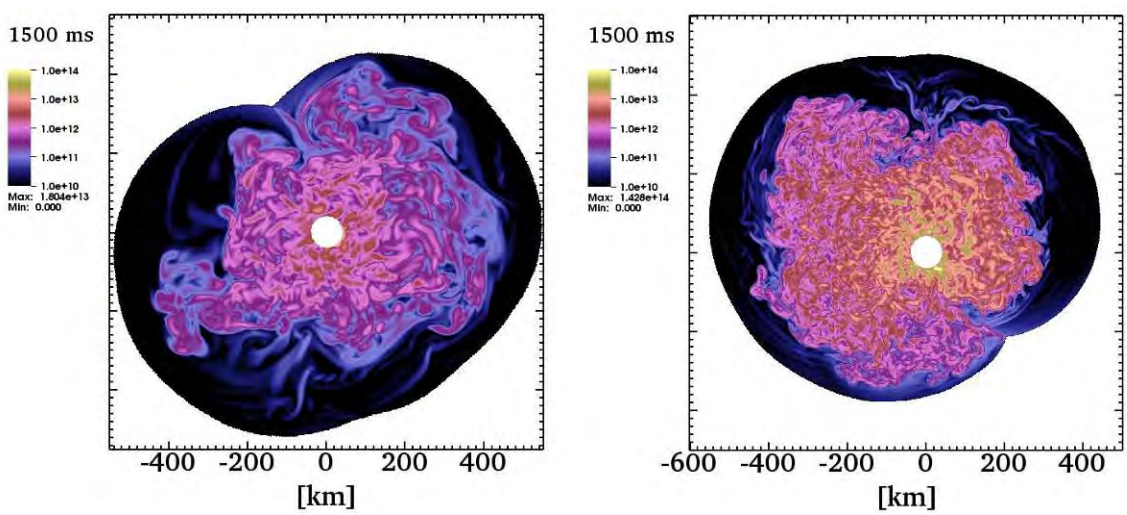


Figure 1. Magnetic field magnitude at t = 1500 ms for models with cell width $\Delta l \approx 2.34$ km (left panel) and $\Delta l \approx 1.17$ km (right panel). The orientation of the plots is chosen so that the normal vector of the slicing plane is parallel to the total angular momentum vector of the flow between the PNS and the shock surface.

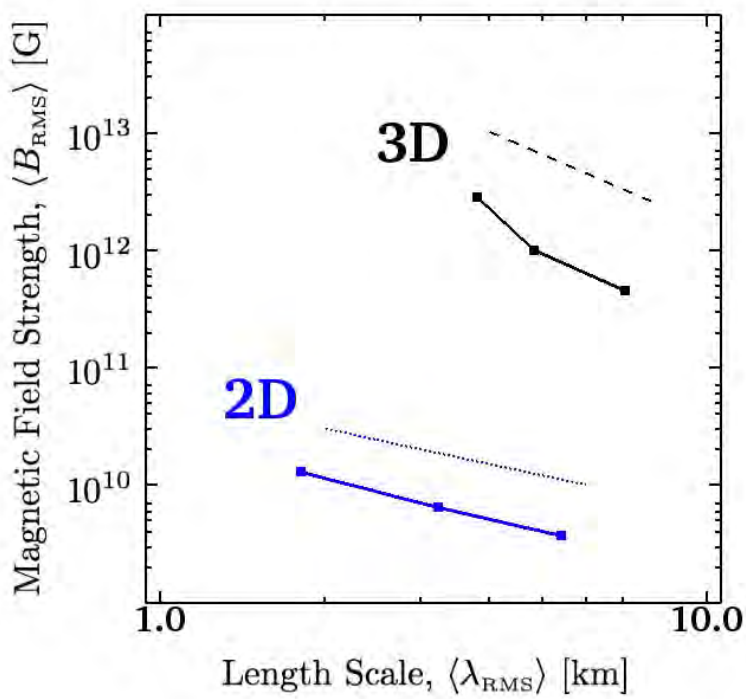


Figure 2. Time-averaged rms magnetic field strength (B_{rms}), versus time-averaged magnetic rms scale (λ_{rms}), in the 'saturated' nonlinear stage. Results are shown (in blue) for the axisymmetric 2D models and (in black) for the 3D models. (The time-average extends from t = 500 ms to the end of each model.) The dotted blue and dashed black reference lines are proportional to (λ_{rms}) and $(\lambda_{rms})^2$, respectively.

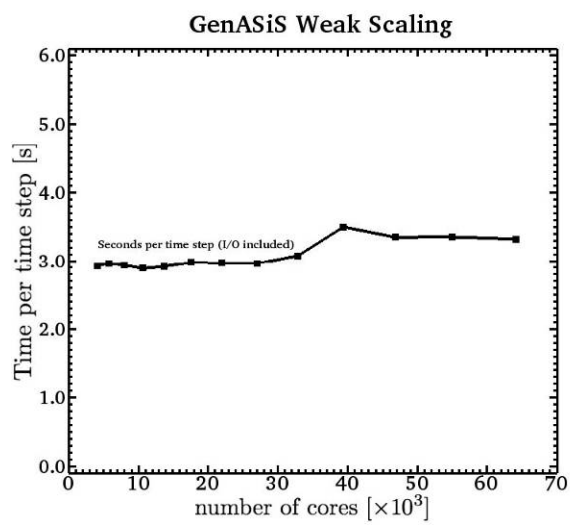


Figure 3. Weak scaling behavior of GenASiS magnetohydrodynamics.

The existence of disparate spatial scales, coupled with the high cost per spatial cell, motivates more efficient methods of crossing spatial scales than simply increasing unigrid resolution. Adaptive mesh refinement (AMR) is one primary means. One possibility is block-structured AMR [11]. In the context of explicit evolution this has the virtue of being able to deploy existing solvers on individual regular cell blocks. There are also efficiencies associated with the use of a predictable basic structure. Another possibility is cell-by-cell AMR with a fully-threaded tree [12]. This offers more fine-grained control over cell division and placement, which is a virtue when the cost per spatial cell is very high. It is not conceptualized around local explicit solvers, so that it might be more amenable to elliptic and implicit solvers which often are needed in a multiphysics context. We are interested in cell-by-cell AMR for GenASiS. Beyond AMR, we also would be interested in subgrid models of turbulence, and in particular turbulent MHD.

Not being aware of a publicly available cell-by-cell refinement library, we undertook to develop this capability on our own. We have found it heavy going for a small application science team, but have made some progress. Figures 4 and 5 illustrate some of our initial efforts at cell-by-cell AMR with GenASiS.

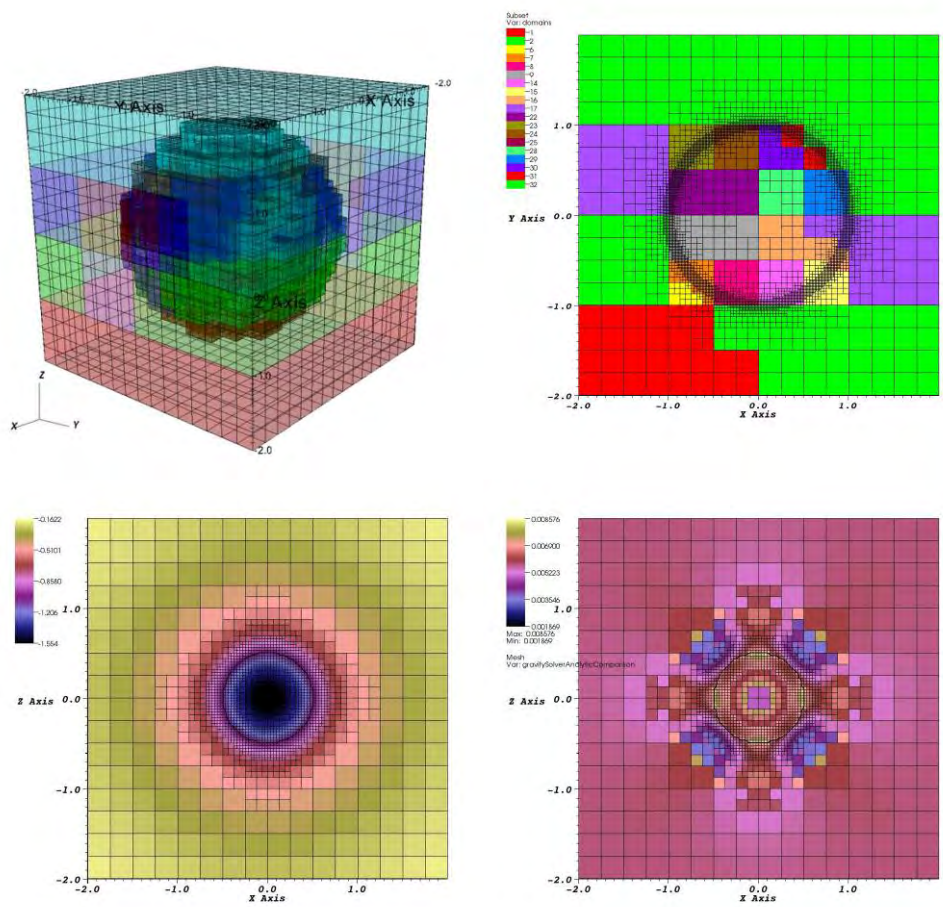


Figure 4. The mesh and its partitioning based on density gradient for a uniform-density sphere; 3D view (upper left) and a 2D slice (upper right). Also shown are the gravitational potential as computed with our multigrid Poisson solver (lower left) and its error relative to the analytic solution (lower right).

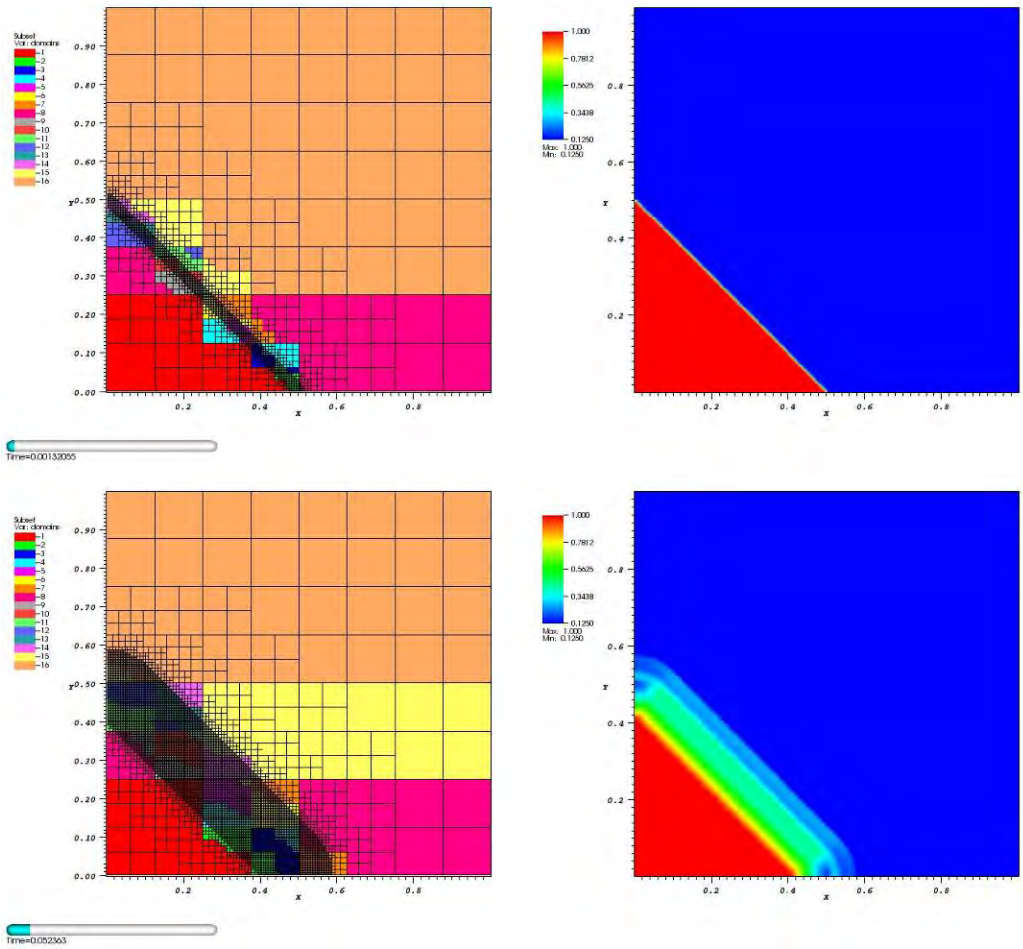


Figure 5. Two snapshots from a 2D shock tube problem illustrate the load rebalancing of our cell-by-cell AMR in a hydrodynamics context. Both the mesh and its partitioning (left panels) and density (right panels) are shown.

We have been discussing spatial scales, but it must also be pointed out that the core-collapse supernova phenomenon is multiscale in time as well. This is handled by implicit evolution of neutrino transport and nuclear reaction kinetics.

4. How Can the Multiphysics Nature of the Problem be Addressed?

From the foregoing sketch of the supernova phenomenon we can see that ideally we would like simulations to cover a wide range of physics, which can be grouped under three main headings: the tangent bundle, the magnetofluid, and the neutrino distributions. The term 'tangent bundle' is a fancy way of saying 'spacetime plus momentum space.' Ideally we would like our representation of spacetime to include all three space dimensions, with good resolution on a wide range of length and time scales; we would like momentum space to include all three dimensions, with good resolution of energies and angles; and ideally self-gravity should be treated with general relativity. The treatment of magnetohydrodynamics must be able to handle shocks. It would be desirable to track nuclear composition using a wide range of species. An equation of state for nuclear matter at finite temperature in neutron-rich conditions is needed. Neutrino transport must be computed in diffusive, decoupling, and free-streaming regimes. Neutrino interactions with all fluid components, and with other neutrinos and antineutrinos, must be included. Neutrino flavor mixing should be included.

Not surprisingly, these different physics pieces involve different types of equations, and correspondingly different kinds of solvers.

In the case of gravity there are some options. If one stays with Newtonian gravity, which is not completely realistic, there is a single linear elliptic equation for the gravitational potential. One can stay with approximate relativity, which captures the enhanced nonlinear strength of relativistic gravity, by solving a set of nonlinear elliptic equations for a reduced set of metric components describing the curvature of spacetime [13]. The in-situ calculation of gravitational waves, however, requires full relativity; the most common formulation these days is as a set of nonlinear first-order hyperbolic equations with continuous solutions satisfying elliptic constraints [14,15]. If elliptic solvers are deployed, multigrid-style approaches seem to be an attractive option. For GenASiS we have worked on one such solver that uses a distributed FFT for the coarsest level and global sparse solves on the finer levels, as illustrated in Figure 4. If only the hyperbolic equations of full relativity are solved a number of methods can be used. High order solvers have been common because of the absence of discontinuities in the gravity variables and the presence of numerical instabilities, but our initial work on full general relativity in GenASiS has been only second order due to the expected availability of only a single layer of ghost cells in our cell-by-cell AMR scheme.

A few different things are required by the magnetohydrodynamics. The main thing is a set of first-order hyperbolic conservation laws—or technically, balance equations, since we have source terms from interactions of the fluid with gravity and neutrinos. The solenoidal constraint on the magnetic field must be respected, in addition to an algebraic constraint known as the equation of state (pressure as a function of density, temperature, and composition). Determination of the equation of state for finite temperature nuclear matter is a significant computational problem in and of itself. A reaction network for time evolution of nuclear abundances is also needed. Many methods have been developed to solve conservation laws. In GenASiS our initial choice is an explicit second-order finite-volume 'central scheme', combined with the 'constrained transport' technique to automatically enforce the solenoidal constraint in the evolution of the magnetic field [16,17]. We have worked with tabulated equations of state for nuclear matter. Local nuclear reaction networks (one for each spatial cell), which must be implicit, have been deployed by our colleagues in CHIMERA but have not yet been put into GenASiS.

For neutrino transport we have first-order hyperbolic integro-differential equations. In conservative form these feature a spacetime divergence, a momentum space divergence, and source terms with integrals over momentum space. Because of the wide range of time scales addressed by transport solvers that range from near-equilibrium diffusive to semi-transparent to nearly free streaming regimes, implicit evolution is a necessity. Past work in our group has used Newton-Raphson to address the nonlinear aspect. Inside each Newton-Raphson iteration is a linear solve. As alluded to previously, this involves dense blocks from momentum space couplings (arising from the integrals over momentum space). It also involves nearest-neighbor spatial couplings, which can be conceptualized as separate sparse matrices, one for every momentum space bin. Our initial strategy in GenASiS is to obtain a large amount of parallelism by separately inverting the dense momentum space blocks and sparse spatial matrices, and iterating these separate solves to fixed-point convergence.

5. How Can the Expected Features of an Exascale Machine be Utilitized?

Some expected features of exascale machines that have penetrated the consciousness of application scientists like ourselves include distributed memory (at least partially), multicore nodes, and heterogeneous processors (use of GPUs and so forth). The first two have been somewhat adapted to using MPI and OpenMP, but our use of these may have to change significantly as communication becomes increasingly expensive. The use of hardware accelerators such as GPUs is something of a brave new world. Nevertheless our colleagues working on CHIMERA have begun experimenting with harnessing GPUs in their implicit solves of nuclear reaction networks.

At some level, awareness of and responsibility for these issues has to percolate all the way up to the application scientists. However we have to rely heavily on, and work together with, applied

mathematicians and computer scientists to help us understand what might be possible and to build libraries that can handle many of these issues in often-needed solvers and kernels. Can what is needed be done at the exascale? Here is a list, with heavy emphasis on expected issues related to data non-locality, of some needs of the several different physics solvers described above:

- Domain decomposition, level-by-level; empirical determination of workload per cell for load balancing? (sparse point-to-point communication for prolongations and restrictions)
- Nearest-neighbor data for explicit hyperbolic equation updates (sparse point-to-point communication overlapping with work)
- Distributed FFT on 3D spatial domain (either parallel FFT, or transposes within slabs)
- Distributed sparse solves over spatial domain (parallel solver libraries)
- Fast inversion of dense blocks (local; use GPUs)
- Fast local sparse spatial solves (local after transpose from space decomposition to momentum space decomposition; use GPUs)
- Residuals, time step determination, global sums (reductions)
- Synchronized evolution of the entire domain in time

On this last point, we are aware that 'synchronization' is a bad word. If one were doing only explicit solves it would be possible to contemplate allowance for asynchrony outside causality cones, but with elliptic and especially implicit solves we have difficulty imagining any way around stepping the entire system forward in time. As far as we can tell, any 'out of order execution' will have to be confined to the solves within a single time step.

As is already known from current experience (but can only be expected to get much worse), collectives—transposes and reductions, in terms of the items on the foregoing list—are the biggest challenge in terms of communication. It may be that non-blocking collectives could open up important new possibilities for overlapping work and communication. To take a very simple example, time step determination could be lagged one or two or a few steps, in order for evolution to continue a bit while a reduction proceeds in the background. Another example particular to radiation transport treating the full momentum space: exposure of an important opportunity for parallelism, namely separate local inversions of a sparse spatial coupling matrix for each momentum space bin, depends on transposing spatial decomposition into momentum space decompositions. Given non-blocking collectives, such a transpose (via All-to-Alls) could occur in the background while the naturally local dense block inversions are performed.

Acknowledgements
We acknowledge computational resources provided through the INCITE program at the Leadership Computing Facility at the National Center for Computational Sciences at Oak Ridge National Laboratory. C. Y. C. and A. M. acknowledge support from the Office of Nuclear Physics, DOE. C. Y. C., E. E., and A. M. acknowledge support from the Office of Advanced Scientific Computing Research, DOE. Oak Ridge National Laboratory is managed by UT-Battelle for the DOE. E. E., R. B., and P. M. acknowledge support from NSF-OCI-0749204. P. M. and A. M. acknowledge support from NASA 07-ATFP07-0011.

References

[1] H. A. Bethe and J. R. Wilson. Revival of a stalled supernova shock by neutrino heating. *Astrophys. J.*, 295:14–23, 1985.
[2] J. M. Blondin, A. Mezzacappa, and C. DeMarino. Stability of standing accretion shocks, with an eye toward core-collapse supernovae. *Astrophys. J.*, 584:971–980, 2003.
[3] J. M. Blondin and A. Mezzacappa. Pulsar spins from an instability in the accretion shock of supernovae. *Nature*, 445:58–60, 2007.
[4] E. Endeve, C. Y. Cardall, R. D. Budiardja, and A. Mezzacappa. Generation of magnetic fields by the stationary accretion shock instability. *Astrophys. J.*, 713:1219–1243, 2010.

[5] S. W. Bruenn, A. Mezzacappa, W. R. Hix, J. M. Blondin, P. Marronetti, O. E. B. Messer, C. J. Dirk, and S. Yoshida. 2d and 3d core-collapse supernovae simulation results obtained with the chimera code. *J. Phys.: Conf. Ser.*, 180, 012018 2009.

[6] K. N Yakunin, P. Marronetti, A. Mezzacappa, S. W Bruenn, C.-T. Lee, M. A Chertkow, W. Hix, J. M Blondin, E. J Lentz, O. Bronson Messer, and S. Yoshida. Gravitational Waves from Core Collapse Supernovae. *Subm. to Class. Quant. Grav.*, arXiv:1005.0779, 2010.

[7] A. Burrows and J. Nordhaus. Core-collapse supernova explosion simulations: The path to and necessity for 3d models. In I. Hubeny, J. M. Stone, K. MacGregor, & K. Werner, editor, *Recent Directions in Astrophysical Quantitative Spectroscopy and Radiation Hydrodynamics: Proceedings of the International Conference in Honor of Dimitri Mihalas for His Lifetime Scientific Contributions on the Occasion of His 70th Birthday*, volume 1171 of American Institute of Physics Conference Series, pages 273–283, 2009.

[8] C. L. Fryer, G. Rockefeller, and M. S. Warren. Snsph: A parallel three-dimensional smoothed particle radiation hydrodynamics code. *Astrophys. J.*, 643:292–305, 2006.

[9] M. Liebendörfer, S. C. Whitehouse, and T. Fischer. The isotropic diffusion source approximation for supernova neutrino transport. *Astrophys. J.*, 698:1174–1190, 2009.

[10] B. Müller, H.-T. Janka, and H. Dimmelmeier. A new multi-dimensional general relativistic neutrino hydrodynamic code for core-collapse supernovae. i. method and code tests in spherical symmetry. *Astrophys. J. Suppl.*, 189:104–133, 2010.

[11] M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, 53:484–512, 1984.

[12] A. Khokhlov. Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations. *J. Comp. Phys.*, 143:519–543, 1998.

[13] Marek, A. and Dimmelmeier, H. and Janka, H.-T. and Müller, E. and Buras, R. Exploring the relativistic regime with newtonian hydrodynamics: an improved effective gravitational potential for supernova simulations. *Astron. Astrophys.*, 445:273–289, 2006.

[14] Shibata, M. and Nakamura, T. Evolution of three-dimensional gravitational waves: Harmonic slicing case. *Phys. Rev. D*, 52:5428–5444, 1995.

[15] Baumgarte, T. W. and Shapiro, S. L. "Numerical integration of Einstein's field equations." *Phys. Rev. D*, 59:024007–+, 1999.

[16] L. Del Zanna and N. Bucciantini. An efficient shock-capturing central-type scheme for multidimensional relativistic flows i. hydrodynamics. *Astron. Astrophys.*, 390:1177–1186, 2002.

[17] L. Del Zanna, N. Bucciantini, and P. Londrillo. An efficient shock-capturing central-type scheme for multidimensional relativistic flows ii. magnetohydrodynamics. *Astron. Astrophys.*, 400:397–413, 2003.

Type Ia supernovae: advances in large scale simulation

H. Ma,[1] M. Zingale,[2] S. E. Woosley,[1] A. J. Aspden,[3] J. B. Bell,[3] A. S. Almgren,[3] A. Nonaka,[3] and S. Dong[1]

[1]Department of Astronomy and Astrophysics, UC Santa Cruz.
[2]Department of Physics and Astronomy, State University of New York, Stony Brook.
[3]CCSE, Lawrence Berkeley National Laboratory.

Abstract. Using the newly developed petascale codes, MAESTRO and CASTRO, the Computational Astrophysics Consortium is simulating the explosion of white dwarf stars as Type Ia supernovae. Since the outcome is sensitive to where the nuclear runaway ignites, three sorts of calculations are being carried out. In the f rst, the presupernova convection of rotating and non-rotating white dwarfs is followed for several hours (star time), using MAESTRO, in order to determine just where the f rst sparks ignite. The turbulent nuclear combustion of the explosion is then followed using the compressible hydro code CASTRO in two other studies that assume either central or off-center ignition. Current calculations are running on 4000 – 12000 CPU, but larger studies, on a greater number of CPU, will be required to increase the Reynolds number of the ignition study and the f delity of the turbulence in the explosion studies.

1. Introduction

Type Ia supernovae (SN Ia) happen when a white dwarf in a binary star system accretes a critical mass and explodes [3]. These are among the brightest explosions in the universe and, because they always starts with the same critical mass, roughly the Chandrasekhar mass, 1.38 solar masses, SN Ia light curves are usually very similar to one another. This makes them excellent "standard candles" for cosmology. There is some diversity however, because not the same fraction of every star burns, and, within that fraction, the composition may vary. Particularly important is the production of radioactive ^{56}Ni, whose decay (6.1 d half ife) to ^{56}Co and ^{56}Co to ^{56}Fe (77.1 d half ife) powers the emissions of the supernova. Indeed, the observed diversity of SN Ia luminosities chief y ref ects their variable ^{56}Ni production. About three-fourths of the iron in nature has been made as ^{56}Ni in these sorts of explosions.

The fraction of the white dwarf that burns depends upon where it is ignited and how fast it burns. If it ignites in the center, then burning proceeds in a large solid angle and the star will become unbound, even without a subsequent detonation. If it ignites off-center, then a buoyant burning bubble of hot ashes f oats off to one side, and only a small fraction of the star is consumed before the burning reaches the surface and is quenched. Such a supernova would either not blow up at all or explode very weakly unless there was a lot of burning after this initial breakout. This fast late time burning must occur as a detonation wave, and the two-stage explosion where this happens is sometimes referred to as a "delayed detonation supernova." A displacement of ignition of only 10–20 km from the center of a star with an initial radius of 1700 km suffices to separate the two classes of models [11]. The actual physics of the detonation transition itself is still uncertain and is debated [6,4,9,2].

2. Ignition

The convective phase of the SN Ia problem is characterized by highly subsonic flow. With a fully compressible hydrodynamics algorithm, the time step would be dictated by the propagation of sound waves, and evolving the system would require a large number of steps and lose accuracy. By filtering out the sound waves while still retaining the compressibility effects important to the flow, the time step is based instead on the fluid velocity itself, giving an enormous efficiency boost. This was the motivation for the development of the low Mach number hydrodynamics code, MAESTRO [5]. With MAESTRO, we performed the first full-star, three-dimensional simulation of the final hours of convection leading up to the ignition of the first flame [10]. Subsequent calculations including rotation (Figure 1) have demonstrated a sensitivity to stellar parameters that may translate into variations in the supernova properties (see also [8]).

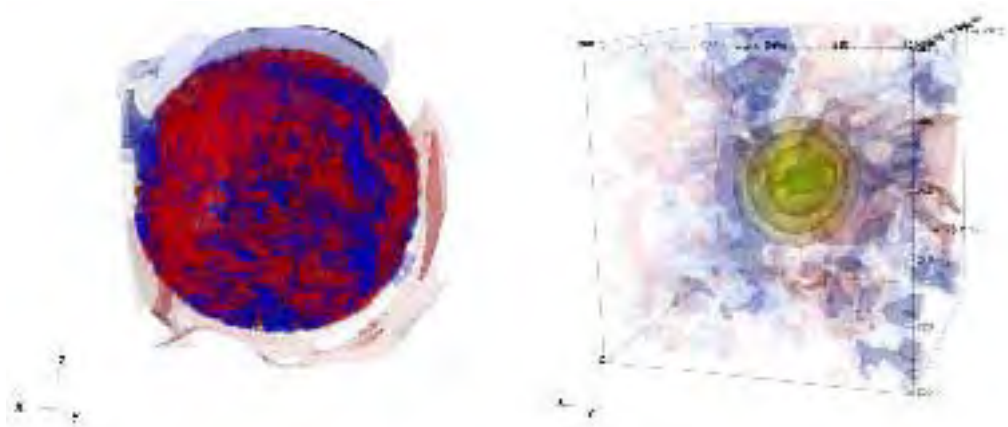


Figure 1. Calculation using MAESTRO of ignition in a rotating Chandrasekhar mass white dwarf shown just seconds before the first flame ignites. In this model, the ratio of centrifugal force to gravity is 1.5% at the equator. The left frame shows radial velocity iso-surfaces. Red indicates outward speeds of 12 (faint red) or 20 km s^{-1} (dark red); blue indicates inward speeds in the same range. The right frame also shows radial velocities with red again indicating motion outwards and blue, inwards. It additionally shows surfaces of constant nuclear energy generation rate (0.32, 1, 3.2, 10, and 32×10^{13} erg g^{-1} s^{-1}) at a time close to the point of run away. Only the inner 1000 km are shown. This calculation required 0.6 M CPU hr on Jaguar at ORNL. Zoning was 3843 which is good for the resources available, but not enough to give a turbulent numerical Reynolds number or to well resolve the ignition radius.

As in past studies, we find that the convective flow demonstrates spontaneous symmetry breaking. An axis for the flow is arbitrarily established that is maintained for a few convective periods, then a rapid migration occurs to another angle. Small amounts of rotation break this pattern. We performed a number of simulations varying the resolution and rotation rate and found ignition radii ranging from essentially central ignition to 85 km off center. This suggests that explosion calculations need to consider both central and strongly off-center ignition.

These calculations are computationally demanding. The calculations using a 384^3 grid required about 0.6 million CPU-hours on 1728 processors to evolve the system for a few hours, and at this resolution, we were still not adequately resolving the turbulence that dominates the convective flow. Our estimates suggest that a resolution of at least 1536^3 will be required both to resolve the inertial range of turbulence and to give an accurate measure (more than a few zones) of the ignition displacement. If uniformly gridded, achieving this higher resolution would increase the necessary computational resources by 4^4 to about 150 M CPU-hours. Keeping work-per-processor constant, such a calculation would run on ~110k processors—MAESTRO is capable of scaling to this regime [1].

Recently however [5], adaptive mesh refinement (AMR) has been added to MAESTRO, resulting in an anticipated savings of about a factor of 6 for this problem and putting the needed resources at 25 M CPU-hours. Even with this savings though, this is a petascale problem.

To date, our calculations end at the ignition of the first flame. However, the ignition process itself may continue, with ignition occurring at multiple points separated closely in time. At the point of the first ignition, the energy release is large enough that the low Mach number approximation breaks down. Over the next year, we will explore ways around this, including modifying the behavior of the reaction network at the point of ignition, adding a simple flame model, and exploring the addition of long-wavelength acoustic information. We have also begun the work necessary to transfer a model [1] from MAESTRO to our compressible hydrodynamics code CASTRO. This can provide another mechanism to explore the potential for subsequent ignition events as well as allow for an exploration of the explosion dynamics with realistic ignition conditions.

3. Explosion

Once born, nuclear burning produces bubbles and sheets of hot ash embedded in cooler, denser fuel. The ash floats, producing Rayleigh-Taylor and Kelvin-Helmholtz instabilities and, ultimately, turbulence. These instabilities, which feed back on the flame sheet, folding and distorting it, govern the rate at which the burning progresses, and, ultimately, the power and brightness of the supernova.

This turbulent nuclear combustion of the explosion was followed using the compressible hydro code CASTRO. Both central and off-center ignition are considered (Figure 2). The initial computational domain was 8,192 km in each dimension, but this was remapped to a larger domain as star expanded. Initially, the base grids of the calculations were 512^3 with 4 levels of AMR and refinement ratios of 2. As the star expanded, the refinement was allowed to decrease. The burning front was modeled as a thick flame and a carbon fraction between 0.1 and 0.4 indicated the location of the flame surface. This and the density were two criteria for AMR refinement. Active flame surface was always tracked at the finest resolution, less than 1 km at the beginning and 4 km at the end, thus the number of finest grids increases rapidly with flame propagation and stellar expansion. Ten species were carried in the calculations, along with two-stage nuclear burning networks calculated both on the flame surface and in the hot ash.

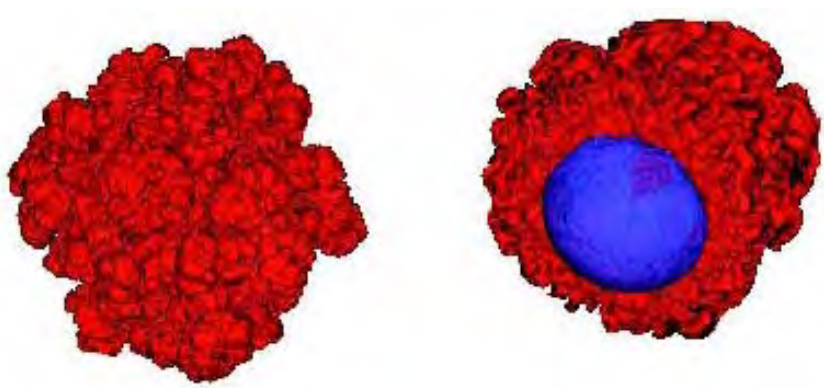


Figure 2. Two 3D simulations of SN Ia explosions using CASTRO. The left figure is of a centrally ignited deflagration with 4 levels of AMR. Red color indicates the flame surface. Initially matter within 100 km was set as hot ash with a perturbed surface. The figure shows the flame surface at a star time of 1 second when the flame has extended to 2500 km away from the center. 0.65 solar masses of iron-group elements have been synthesized. The right frame shows the results of a study of off-center ignition at a time just before the collision of ejecta on the far side, and blue color indicates the star surface. This run has a resolution of 5 km around the flame surface. Only about 0.08 solar masses of iron group elements were produced prior to the collision.

Because of its larger solid angle the central ignition case was more costly. Our first calculation ran on 4096 to 8192 CPU on Franklin and 12288 CPU on Jaguar for about 1.9 MCPU hr. Even though we dropped the finest resolution to 2 km then to 4 km, the total number of cells exceeded 1.5 billion. At a star time near 1.5 second, nuclear burning was essentially over and 0.70 solar masses of iron-group elements and 0.1 solar masses of intermediate elements (Si, S, Ar, Ca) had been produced. The star was exploding with a projected final kinetic energy of 8×10^{50} erg. This energy and iron mass are comparable to what was obtained in a previous study by the Munich group [7], but the synthesis of intermediate mass elements here is much lower. Our calculation needs to be done with greater resolution in order to reduce sensitivity to the flame model, but the Munich result for low density burning may be questionable. A calculation that fully resolved the turbulent integral scale, about 10 km, during the critical late stages of the explosion would require about 40 MCPU hr and run on 125 k CPU.

The off-center case, on the other hand, sends up a burning vortex ring to one side whose ashes erupt and roll around the star. We are looking to see whether a detonation will develop when a collision occurs on the far side of the star, as has been suggested by the Chicago FLASH team. We have studied this problem in 2D and 3D for two different choices of resolution, but are still in an exploratory phase using only moderate resolution. Our 2D results agree with both Chicago and Munich in that only a small amount of white dwarf expansion occurs during the initial rise of the bubble to the surface and spread to the far side. A vigorous collision could ignite a detonation. Our 3D calculations, however show more burning and a result that is very sensitive to just where the ignition occurs. For the small ignition radius implied by the MAESTRO studies just discussed (~30 km), the collision on the far side of the star may be weaker than Chicago has estimated. Typical burned masses before the collision happens are 0.08 $M_\odot$, not enough to unbind the star, but enough to cause significant expansion. However, we need to do our calculations at higher resolution and study a variety of ignition radii before drawing strong conclusions here. Fortunately, given the computer resources, we can do that. A definitive study that resolved the turbulence during the deflagration stage would take about 20 MCPU hr on 80 k CPU.

Because the flame surface tracked at the finest level occupies only a small fraction of the star, a typical simulation of the pre-detonation phase here spans 1.3 seconds of star time and has used only 0.4 MCPU hr, i.e., five times less than the central ignition study. Multiple time steps and AMR speed up the study a lot. The resolution at the flame early in the study was less than 1 km and was 5 km at late times. We plan to increase the late time resolution by a factor of at least two in the near future and to finely resolve the collision of the ashes on the far side of the white dwarf.

Acknowledgements

Research at LBNL was supported by the SciDAC Program under contract No. DE-AC0205CH11231. A. Aspden was also supported by a Seaborg Fellowship. Work at UCSC was also supported by SciDAC under contract DE-FC02-06ER41438. M. Zingale was supported by a DOE/Office of Nuclear Physics Outstanding Junior Investigator award, grant No. DE-FG0206ER41448, to Stony Brook. This research used resources provided by an INCITE award at the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the DOE (DE-AC05-00OR22725), and the Franklin computer at the National Energy Research Scientific Computing Center. Additional computing resources were provided on the ATLAS Linux Cluster at LLNL as part of a Grand Challenge Project.

References

[1] A. Almgren, J. Bell, D. Kasen, M. Lijewski, A. Nonaka, P. Nugent, C. Rendleman, R. Thomas, and M. Zingale. MAESTRO, CASTRO, and SEDONA—Petascale Codes for Astrophysical Applications. 2010. in this proceedings.

[2] A. J. Aspden, J. B. Bell, and S. E. Woosley. Distributed Flames in Type Ia Supernovae. *The Astrophysical Journal*, 710:1654–1663, February 2010.

[3] F. Hoyle and W. A. Fowler. Nucleosynthesis in Supernovae. *The Astrophysical Journal*, 132:565–590, November 1960.

[4] G. C. Jordan, IV, R. T. Fisher, D. M. Townsley, A. C. Calder, C. Graziani, S. Asida, D. Q. Lamb, and J. W. Truran. Three-Dimensional Simulations of the Def agration Phase of the Gravitationally Conf ned Detonation Model of Type Ia Supernovae. *The Astrophysical Journal*, 681:1448–1457, July 2008.

[5] A. Nonaka, A. S. Almgren, J. B. Bell, M. J. Lijewski, C. M. Malone, and M. Zingale. MAESTRO: An adaptive low mach number hydrodynamics algorithm for stellar f ows. *The Astrophysical Journal Supplement Series*, 188:358–383, 2010.

[6] F. K. Röpke. Flame-driven Def agration-to-Detonation Transitions in Type Ia Supernovae? *The Astrophysical Journal*, 668:1103–1108, October 2007.

[7] F. K. Röpke, W. Hillebrandt, W. Schmidt, J. C. Niemeyer, S. I. Blinnikov, and P. A. Mazzali. A Three-Dimensional Def agration Model for Type Ia Supernovae Compared with Observations. *The Astrophysical Journal*, 668:1132–1139, October 2007.

[8] S. E. Woosley, A. Almgren, J. B. Bell, G. Glatzmaier, D. Kasen, A. R. Kerstein, H. Ma, P. Nugent, F. Röpke, V. Sankaran, and M. Zingale. Type Ia supernovae. *Journal of Physics Conference Series*, 78(1):012081–+, July 2007.

[9] S. E. Woosley, A. R. Kerstein, V. Sankaran, A. J. Aspden, and F. K. Röpke. Type Ia Supernovae: Calculations of Turbulent Flames Using the Linear Eddy Model. *The Astrophysical Journal*, 704:255–273, October 2009.

[10] M. Zingale, A. S. Almgren, J. B. Bell, A. Nonaka, and S. E. Woosley. Low Mach Number Modeling of Type IA Supernovae. IV. White Dwarf Convection. *The Astrophysical Journal*, 704:196–210, October 2009.

[11] M. Zingale and L. J. Dursi. Propagation of the First Flames in Type Ia Supernovae. *The Astrophysical Journal*, 656:333–346, February 2007.

Hybrid parallelism for volume rendering on large, multi-core systems

Mark Howison, E. Wes Bethel, Hank Childs

Visualization Group, Lawrence Berkeley National Laboratory

Email: {mhowison,ewbethel,hchilds}@lbl.gov

Abstract. This work studies the performance and scalability characteristics of "hybrid" parallel programming and execution as applied to raycasting volume rendering—a staple visualization algorithm—on a large, multi-core platform. Historically, the Message Passing Interface (MPI) has become the de-facto standard for parallel programming and execution on modern parallel systems. As the computing industry trends towards multi-core processors, with four- and six-core chips common today and 128-core chips coming soon, we wish to better understand how algorithmic and parallel programming choices impact performance and scalability on large, distributed-memory multi-core systems. Our findings indicate that the hybrid-parallel implementation, at levels of concurrency ranging from 1,728 to 216,000, performs better, uses a smaller absolute memory footprint, and consumes less communication bandwidth than the traditional, MPI-only implementation.

1. Introduction

It is well accepted that the future of parallel computing involves chips that are comprised of many (smaller) cores. With this trend towards more cores on a chip, many in the HPC community have expressed concern that parallel programming languages, models, and execution frameworks that have worked well to-date on single-core massively parallel systems may "face diminishing returns" as the number of computing cores on a chip increase [1]. In this context, we explore the performance and scalability of a common visualization algorithm—raycasting volume rendering—implemented with different parallel programming models and run on a large supercomputer comprised of six-core chips. We compare a traditional implementation based on message-passing against a "hybrid" parallel implementation, which uses a blend of traditional message-passing (inter-chip) and shared-memory (intra-chip) parallelism (Figure 1).

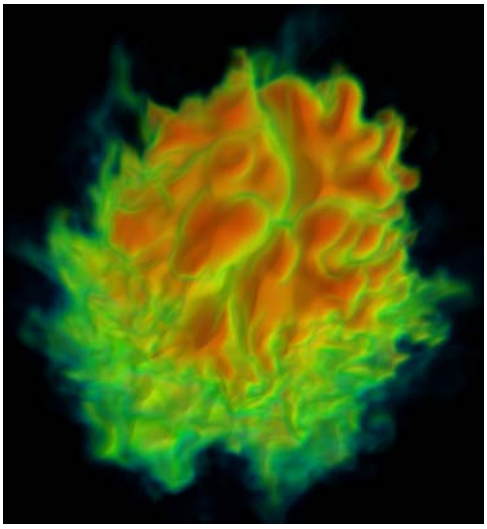


Figure 1. This 4608^2 image of a combustion simulation result was rendered by our MPI+pthreads implementation running on 216,000 cores of the JaguarPF supercomputer.

Over the years, there has been a consistent and well-documented concern that the overall runtime of

large-data visualization algorithms is dominated by I/O costs (e.g., [2,3]). During our experiments, we observed results consistent with previous work: there is a significant cost associated with scientific data I/O. In this study, however, we focus exclusively on the performance and scalability of the ray casting volume rendering algorithm, not on parallel I/O performance. This approach is valid for many visualization use cases, such as creating multiple images from a single dataset that fits entirely within the memory footprint of a large system, or creating one or more images of data that is already resident in memory, as in the case of *in-situ* visualization.

Our findings show that there is indeed opportunity for performance gains when using hybrid-parallelism for raycasting volume rendering across a wide range of concurrency levels. The hybrid-parallel implementation runs faster, requires less memory, and for our particular algorithm and set of implementation choices, consumes less communication bandwidth than the traditional, MPI-only implementation.

2. Implementation

Our implementation of a parallel raycasting volume renderer draws on a large body of prior research (see [4] for a detailed overview). The primary contributions of our work is a comparison of the performance and resource requirements between hybrid- and traditional-parallel programming models.

From a high level view, our parallel volume renderer distributes a source data volume S across n parallel MPI PEs. Each PE reads in $\frac{1}{n}$ of S, performs raycasting volume rendering on this data subdomain to produce a set of image fragments, then participates in a compositing stage in which fragments are exchanged and combined into a final image. The completed image is gathered to PE 0 for display or I/O to storage. Our distributed memory parallel implementation is written in C++ and C and makes calls to MPI. The portions of the implementation that are shared-memory parallel are written using a combination of C++ and C and either POSIX threads or OpenMP, so that we are comparing two hybrid implementations, MPI+pthreads and MPI+OpenMP.

Our raycasting volume rendering code implements Levoy's method [5]: we compute the intersection of a ray with the data block, and then compute the color at a fixed step size along the ray through the volume. All such colors along the ray are integrated front-to-back using the "over" operator. Output from our algorithm consists of image fragments that contain an x, y pixel location, R, G, B, α color information, and a z-coordinate. The z-coordinate is the location in eye coordinates where the ray penetrates the block of data. In the MPI-only case, this serial implementation is invoked on each of the PEs. Each operates on its own disjoint block of data. As we are processing structured rectilinear grids, all data subdomains are spatially disjoint, so we can safely use the ray's entry point into the data block as the z-coordinate for sorting during the later composition step.

In the MPI-hybrid case, the raycasting volume renderer on each MPI PE is a shared-memory parallel implementation consisting of T threads all executing concurrently to perform the raycasting on a single block of data. As in [6], we use an image-space partitioning: each thread is responsible for raycasting a portion of the image. Our image-space partitioning is interleaved, where the image is divided into many small tiles that are distributed amongst the threads. Through a process of manual experimentation, we determined that an image tile size of 32 × 32 pixels produced consistently better performance than other tile sizes for a variety of source volume sizes on the six-core AMD Opteron processor. We also found that a dynamic work assignment of tiles to threads minimized load imbalance.

Compositing begins by partitioning the pixels of the final image amongst the PEs. Next, an all-to-all communication step exchanges each fragment from the PE where it was generated in the raycasting phase to the PE that owns the region of the image in which the fragment lies. This exchange is implemented using an MPI_Alltoallv call, which provides the same functionality as direct sends and receives, but bundles the messages into fewer point-to-point exchanges for greater efficiency. Each PE then performs the final compositing for each pixel in its region of the image using the "over" operator, and the final image is gathered on PE 0.

Peterka et al. [3] reported scaling difficulties for compositing when using more than 8,000 PEs. They solved this problem by reducing the number of PEs receiving fragments to be no more than 2,000. We

emulated this approach, again limiting the number of PEs receiving fragments, although we experimented with values higher than 2,000.

In the hybrid implementations, only one thread per socket participates in the compositing phase. That thread gathers fragments from all other threads in the same socket, packs them into a single buffer, and transmits them to other compositing PEs. This approach results in fewer messages than if all threads in the hybrid parallel implementation were to send messages to all other threads on all other CPUs. Our aim here is to better understand the opportunities for improving performance in the hybrid-parallel implementation. The overall effect of this design choice is an improvement in communication characteristics, as indicated in Section 3.3.

3. Results

We conducted a strong scaling study where we rendered a 4608^2 image from a 4608^3 dataset (roughly 97.8 billion cells) at concurrency levels from 1,728-way parallel to 216,000-way parallel. Our test system, JaguarPF, is a Cray XT5 located at Oak Ridge National Laboratory. Each of its 18,688 nodes has two sockets, and each socket has a six-core 2.6GHz AMD Opteron processor, for a total of 224,256 compute cores. With 16GB per node (8GB per socket), the system has 292TB of aggregate memory and 1.3GB per core.

On JaguarPF, a minimum of 1,728 cores is required to accommodate this particular problem size. In the hybrid case, we shared a data block among six threads and used one sixth as many MPI PEs. Although we could have shared a data block among as many as twelve threads on each dual-socket six-core node, we chose not to because sharing data across sockets results in non-uniform memory access. Based on preliminary tests, we estimated this penalty to be around 5 or 10% of the raycasting time. Therefore, we used six threads running on the cores of a single six-core processor. Because the time to render is view-dependent, we executed each raycasting phase ten times over a selection of ten different camera locations. The raycasting times we report are an average over all locations.

In the compositing phase, we tested five different ratios of total PEs to compositing PEs. We restricted the compositing experiment to only two views (the first and last) because it was too costly to run a complete battery of all view and ratio permutations. Since the runtime of each trial can vary due to contention for JaguarPF's interconnection fabric with other users, we ran the compositing phase ten times for both views. We report mean and minimum times over this set of twenty trials. Minimum times most accurately represent what the system is capable of under optimal, contention-free conditions, while mean times characterize the variability of the trials.

Starting with a 512^3 dataset of combustion simulation results,* we used trilinear interpolation to upscale it to arbitrary sizes in memory. We scaled equally in all three dimensions to maintain a cubic volume. Our goal was to choose a problem size that came close to filling all available memory. Although upscaling may distort the results for a data-dependent algorithm, the only data dependency during raycasting is early ray termination. However, we found that for our particular dataset and transfer function, there was always at least one data block for which no early terminations occurred. Moreover, the cost of the extra conditional statement inside the ray integration loop to test for early termination added a 5% overhead. Therefore, we ran our study with early ray termination turned off, and we believe that upscaling the dataset does not effect our results.

Because the compute nodes on an XT5 system have no physical disk for swap, and hence no virtual memory, exceeding the physical amount of memory causes program termination. Our total memory footprint was four times the number of bytes in the entire dataset: one for the dataset itself, and the other three for the gradient data, which we computed by central difference and used in shading calculations. Although each node has 16GB of memory, we could reliably allocate only 10.4GB for the data block and gradient field at 1,728-way concurrency because of overhead from the operating system and MPI runtime. We chose concurrencies that are cubic numbers to allow for a clean decomposition of the entire volume into cubic blocks per MPI PE.

* Sample data courtesy J. Bell and M. Day, Center for Computational Sciences and Engineering, LBNL.

3.1. Memory Usage

Because MPI-hybrid uses fewer MPI PEs, it incurs less memory overhead from the MPI runtime environment and from program-specific data structures that are allocated per PE. We measured the memory footprint of the program directly after calling MPI_Init using the VmRSS, or "resident set size," value from the /proc/self/status interface. Memory usage was sampled only from MPI PEs 0 through 6, but those values agreed within 1–2%. At 216,000 cores, the per-PE runtime overhead of MPI-only was 176MB, more than twice the 82MB required by MPI-hybrid. The MPI-only per-node (2,106MB) and aggregate (37,023MB) memory usage was another factor of six larger than MPI-hybrid (165MB per-node, 2,892MB aggregate) because it uses 6× as many PEs as MPI-hybrid. Thus, MPI-only used nearly 12× as much memory per-node and in-aggregate than MPI-hybrid to initialize the MPI runtime at 216,000-way concurrency.

Two layers of ghost data are required in our raycasting phase: the first layer for trilinear interpolation of sampled values, and the second layer for computing the gradient field using central differences (gradients are not precomputed for our dataset). Because the MPI-hybrid approach uses fewer, larger blocks in its decomposition, it requires less exchange and storage of ghost data by roughly 40% across all concurrencies.

3.2. Raycasting

Our raycasting phase scales nearly linearly because it involves no inter-processor communication (see Figure 2). Each MPI PE obtains its data block, then launches either one (MPI-only) or six (MPI-hybrid) raycasting threads. Working independently, each thread tests for ray-triangle intersections along the data block's bounding box and, in the case of a hit, integrates the ray by sampling data values at a fixed interval along the ray, applying a transfer function to the values, and aggregating the resulting color and opacity in a fragment for that ray's image position. For these runs and timings, we use trilinear interpolation for data sampling along the ray as well as a Phong-style shader. The final raycasting time is essentially the runtime of the thread that takes the most integration steps, and this behavior is entirely dependent on the view.

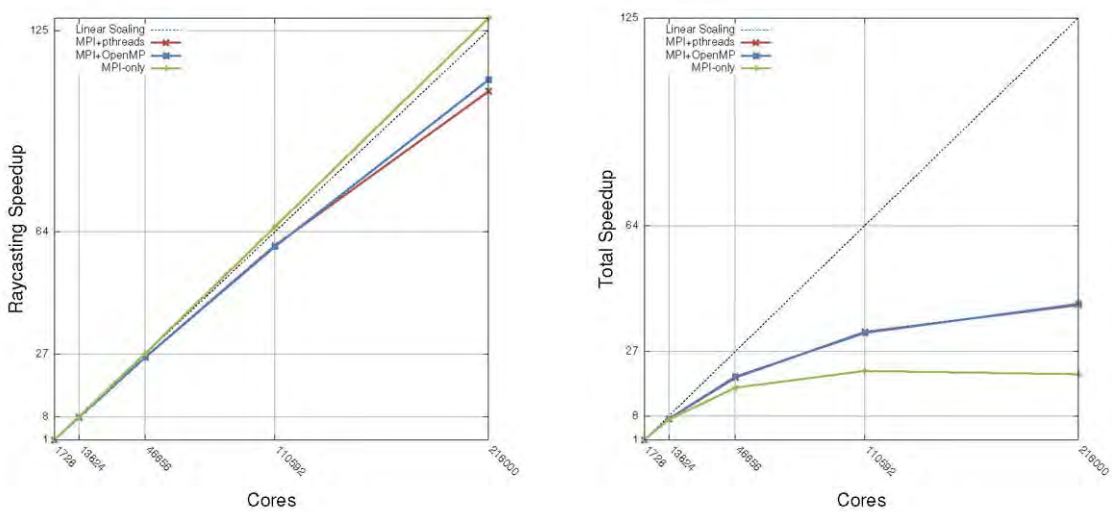


Figure 2. The speedups (referenced to 1,728 cores) for both the raycasting phase and the total render time (raycasting and compositing).

Overall, we have achieved linear scaling up to 216,000 for the raycasting phase with MPI-only (see Figure 2). MPI-hybrid exhibits different scaling behavior because it has a different decomposition geometry: MPI-only has a perfectly cubic decomposition, while MPI-hybrid aggregates 1 × 2 × 3 cubic blocks into rectangular blocks that are longest in the *z*-direction. The interaction of the decomposition

geometry and the camera direction determine the maximum number of ray integration steps, which is the limiting factor for the raycasting time. At lower concurrencies, this interaction benefits MPI-hybrid, which outperforms MPI-only by as much as 11%. At higher concurrencies the trend flips, and MPI-only outperforms MPI-hybrid by 10%. We expect that if we were able to run on an eight-core system with a $2 \times 2 \times 2$ aggregation factor for MPI-hybrid, both implementations would scale identically. We also note that at 216,000 cores, raycasting is less than 20% of the total runtime (see Figure 4), and MPI-hybrid is over 50% faster because of gains in the compositing phase that we describe next.

3.3. Compositing

We observe that with increasing concurrency, the MPI-hybrid compositing times are systematically better than the MPI-only implementation. The primary cost of compositing is the MPI_Alltoallv call that moves each fragment from the PE where it originated during raycasting to the compositing PE that owns the region of image space where the fragment lies. Because MPI-hybrid aggregates these fragments in the memory shared by six threads, it uses on average about 6× fewer messages than MPI-only (see Figure 3). In addition, MPI-hybrid exchanges less fragment data because its larger data blocks allow for more compositing to take place during ray integration.

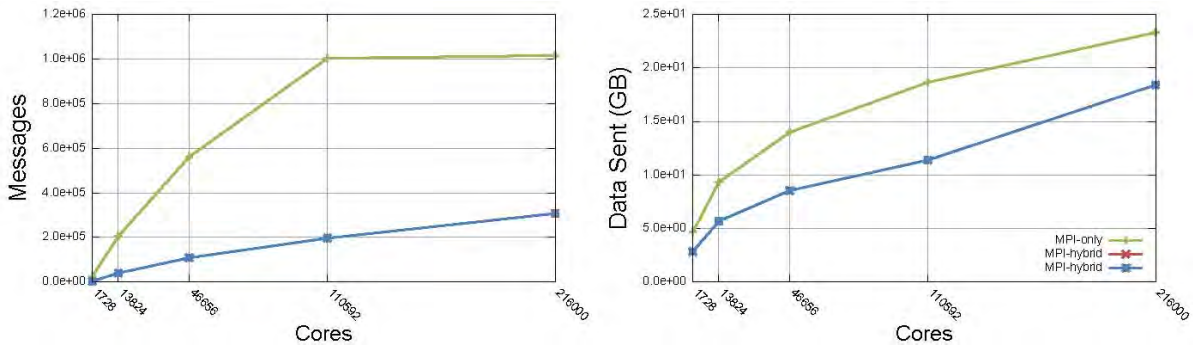


Figure 3. The number of messages and total data sent during the compositing fragment exchange.

Overall, the best compositing time at 216,000 cores for MPI-hybrid (0.35s, 4500 compositors) is 67% less than for MPI-only (1.06s, 6750 compositors) because the MPI-hybrid approach results in fewer and smaller messages than the MPI-only implementation. Furthermore, at this scale compositing time dominates rendering time, which is roughly 0.2s for both MPI-only and MPI-hybrid. Thus, the total render time is 55% faster for MPI-hybrid (0.56s versus 1.25s). The advantage of MPI-hybrid over MPI-only also becomes greater as the number of cores increases (see Figure 4).

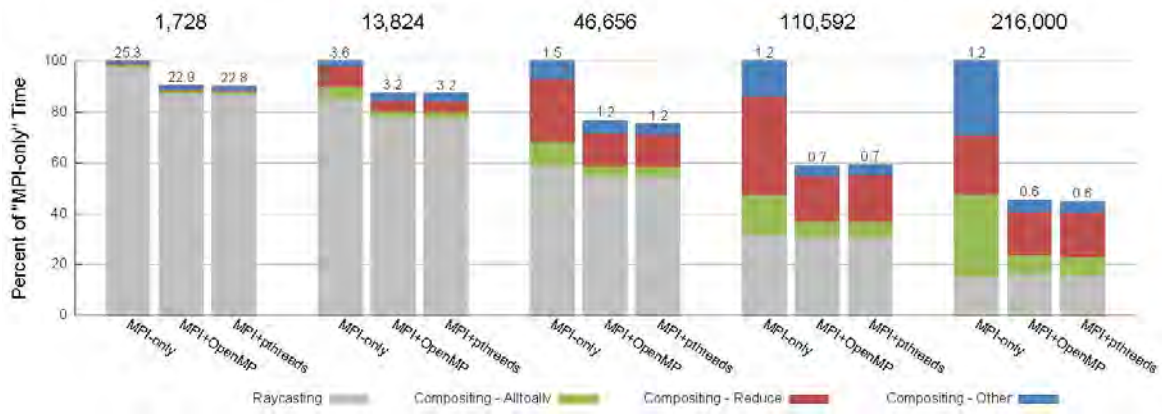


Figure 4. In terms of total render time, MPI-hybrid outperforms MPI-only at every concurrency level, with performance gains improving at higher concurrency.

4. Conclusion and Future Work

The multi-core era offers new opportunities and challenges for parallel applications. This study has shown that hybrid parallelism offers performance improvements and better resource utilization for raycasting volume rendering on a large, distributed-memory supercomputer comprised of multi-core CPUs. The advantages we observe for hybrid parallelism are reduced memory footprint, reduced MPI overhead, and reduced communication traffic. These advantages are likely to become more pronounced in the future as the number of cores per CPU increases while per-core memory size and bandwidth decrease.

We found that at high concurrency, fragment exchange during the compositing phase is the most expensive operation. Our compositing algorithm relies entirely on the implementation of the MPI_Alltoallv call in the Cray MPI library. Our finding that using a subset of compositing PEs reduces communication overhead agrees with what Peterka et al. [3] found for the MPI implementation on an IBM BG/P system, suggesting that both libraries use similar implementations and optimizations of MPI_Alltoallv. A separate paper by Peterka et al. [7] introduces a new compositing algorithm, "Radix-k," that shows good scaling up to 16,000-way concurrency. This approach, which is compatible with our hybrid parallel system, may lead to even faster compositing times. Studying their combined performance, especially at concurrencies in the hundreds of thousands, is an avenue for future work.

Acknowledgment

This work was supported by the Director, Office of Science, Office and Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET). This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Additionally, preliminary results for this work were collected using resources of the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

[1] Asanovic, K., et al. (2006). The landscape of parallel computing research: A view from Berkeley. Tech. Rep. UCB/EECS-2006-183, EECS Department, University of California, Berkeley.

[2] Kniss, J., McCormick, P., McPherson, A., Ahrens, J., Painter, J., Keahey, A., and Hansen, C. (2001). Interactive texture-based volume rendering for large data sets. *IEEE Computer Graphics and Applications*, **21**, 52–61.

[3] Peterka, T., Yu, H., Ross, R., Ma, K.-L., and Latham, R. (2009). End-to-end study of parallel volume rendering on the IBM Blue Gene/P. *ICPP '09: Proceedings of the 2009 International Conference on Parallel Processing*, Washington, D.C., USA, pp. 566–573, IEEE Computer Society.

[4] Howison, M., Bethel, E. W., and Childs, H. (2010). MPI-hybrid parallelism for volume rendering on large, multi-core systems. *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, Norrköping, Sweden, May, lBNL-3297E.

[5] Levoy, M. (1988). Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, **8**, 29–37.

[6] Nieh, J. and Levoy, M. (1992). Volume rendering on scalable shared-memory MIMD architectures. *Proceedings of the 1992 Workshop on Volume Visualization*, October, pp. 17–24, ACM SIGGRAPH.

[7] Peterka, T., Goodell, D., Ross, R., Shen, H.-W., and Thakur, R. (2009). A configurable algorithm for parallel image-compositing applications. *SC '09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, New York, NY, USA, pp. 1–10, ACM.

Performance of communication patterns for extreme-scale analysis and visualization

T Peterka,[1] W Kendall,[2] D Goodell,[1] B Nouanesengsey,[3] H-W Shen,[3] J Huang,[2] K Moreland,[4] R Thakur,[1] and R B Ross[1]

[1]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA
[2]Department of Electrical Engineering and Computer Science, University of Tennessee at Knoxville, Knoxville, TN 37996, USA
[3]Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210, USA
[4]Sandia National Laboratories, Albuquerque, NM 87185, USA

Email: tpeterka@mcs.anl.gov

Abstract. Efficient data movement is essential for extreme-scale parallel visualization and analysis algorithms. In this research, we benchmark and optimize the performance of collective and point-to-point communication patterns for data-parallel visualization of scalar and vector data. Two such communication patterns are global reduction and local nearest-neighbor communication. We implement scalable algorithms at tens of thousands of processes, in some cases to the full scale of leadership computing facilities, and benchmark performance using large-scale scientific data.

1. Introduction

Through funding initiatives such as SciDAC, the U.S. Department of Energy's Office of Science has embarked on a historic path in computational and computer science. At tera- and petascale today, and certainly at exascale in the future, data analysis and visualization will continue to be pivotal components in the success of this endeavor [1]. Efficient data movement, whether storage access or network communication, is essential for the success of parallel analysis algorithms at extreme scale.

This paper summarizes the performance of two network communication patterns for parallel analysis and visualization: global reduction and local nearest-neighbor communication. Global reduction is collective communication whereby all processes participate in merging their results into one solution. Local nearest-neighbor is an assortment of sparse collective neighborhoods, and each process communicates with only those processes in its immediate vicinity. Case studies of the scalability of the Radix-k algorithm for parallel image compositing and of parallel particle tracing are presented as examples of each communication model. The two case studies represent high impact long-standing scalability challenges with the potential to impede progress in extreme-scale visualization; one is a critical bottleneck in rendering while the other is a first step in many flow visualization and feature extraction algorithms.

2. Algorithms and Data Structures for Communication in Visualization

Sort-last parallel visualization algorithms are data-parallel approaches that partition the data space among processes and execute the same visualization task concurrently on each process. Upon completion of this step, each process owns an output image that must be merged with all the other processes' images to form a single result. Image composition is the name given to this stage, and it relies on an efficient global reduction communication pattern. Previously, direct-send [2] and binary swap [3] were the accepted best practices for performing image composition, although recently Radix-k [4,5] has demonstrated significant performance gains on a variety of machine architectures.

Radix-k enables the amount of communication concurrency to be tuned to the architecture by factoring the number of processes into a number of rounds and a number of communicating partners in a group in each round (see Figure 1). By configuring the k-values (group size in each round) appropriately, the available network bisection bandwidth can be approached without exceeding it and generating contention for messages. With Radix-k, direct-send and binary swap simply become two of the numerous valid configurations possible for a given number of processes. Radix-k also overlaps the communication of messages with the reduction computation as much as possible, further improving performance.

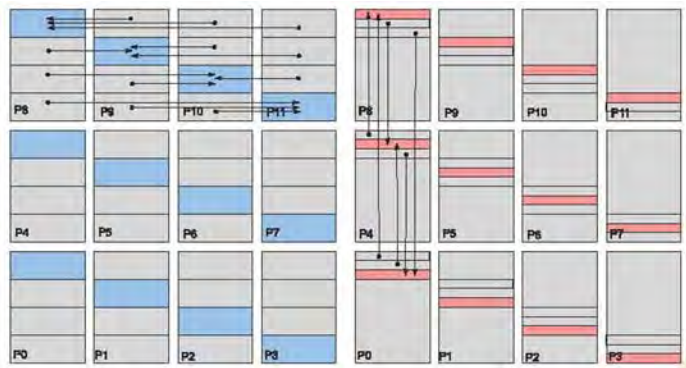


Figure 1. Illustration of the Radix-k parallel image compositing algorithm on 12 processes. Any factorization of the total number of processes is a valid configuration; in this example, two rounds of [4,3] are used. Groups of 4 are formed in the first round (left), and groups of 3 are formed in the second round (right). Within each group, a local direct-send pattern is executed.

Some analysis and visualization applications require a local message exchange among neighbors instead of global reduction. This is the case when particles are advected by a flow field for computing streamlines or pathlines. As each particle crosses block boundaries, it is relayed to the process that owns the neighboring block. Or, as Pugmire et al. [6] demonstrated, a new data block can be loaded by the same process as an alternative to handing the particle to another process. In our algorithm, the message sizes involved in communicating particles are smaller than the data movement required to load another data block from storage. Thus, we use a static block distribution and focus our efforts instead on efficient communication.

Figure 2 shows the basic block structure used in our nearest-neighbor communication. The dataset consists of

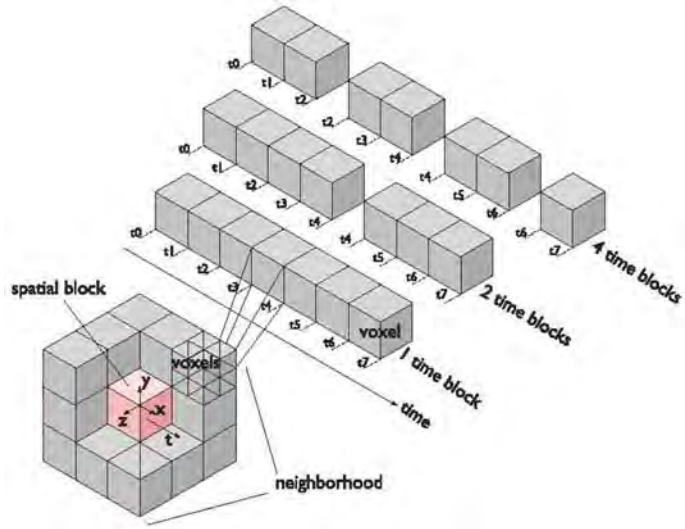


Figure 2. The data structures for nearest-neighbor communication of 4D particles are overlapping neighborhoods of 81 4D blocks. Each block consists of voxels and has extents in the (x, y, z, t) dimensions. In the example above, eight time-steps are grouped into either one, two, or four blocks in the time dimension.

multiple time-steps, one per file, containing 3D velocity vectors; and because we accommodate time-varying datasets and unsteady flow fields, all particles and blocks are 4D entities. A neighborhood consists of a central block surrounded by 80 other neighbors, for a total neighborhood size of 81 blocks. That is, the neighborhood is a 3 × 3 × 3 × 3 region comprising the central block and any other block adjacent in space and time. These neighborhoods are partially overlapping; hence, the neighbor relation is reflexive and symmetric but not transitive. A particle is transferred from one block to another within a neighborhood whenever one or more particle coordinates (x, y, z, t) exceed the block boundary in any of the four dimensions.

3. Performance Results

We tested Radix-k image compositing and parallel particle tracing on some of the largest supercomputers in the world and present selected results below. The Argonne Leadership Computing Facility has the IBM Blue Gene/P (BG/P) *Intrepid* system while the Oak Ridge National Center for Computational Sciences maintains the Cray XT5 *Jaguar* system. Intrepid has 160 K cores while Jaguar has 219 K cores; Jaguar is currently the fastest supercomputer in the world according to the June 2010 Top 500 listing, while Intrepid is currently ranked ninth.

Figure 3 shows the results of testing Radix-k in a volume rendering application applied to a core-collapse supernova dataset on both Intrepid and Jaguar. In this research, bounding box and run-length encoding optimizations were implemented in Radix-k, and the compositing algorithm was inserted in our parallel volume rendering code [7].

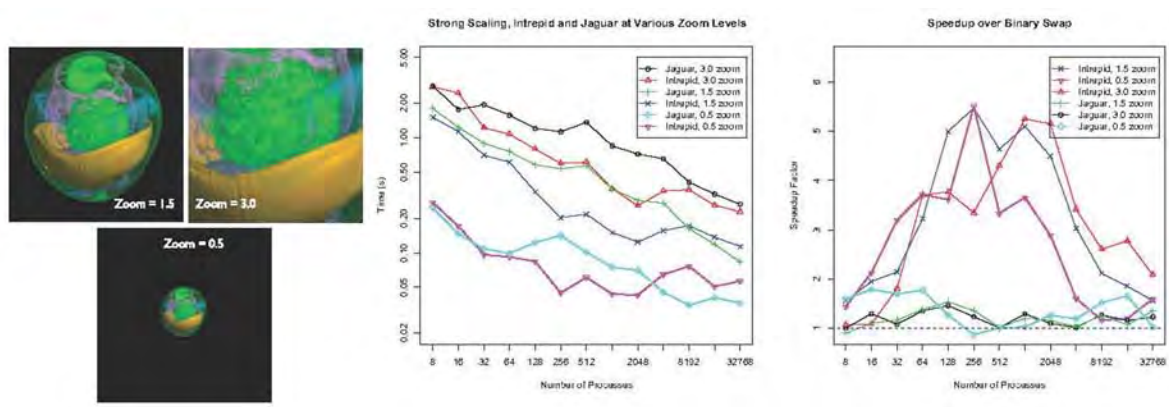


Figure 3. When optimizations such as bounding boxes and run-length encoding are implemented in Radix-k for the volume rendering of core-collapse supernovae (left), the compositing time (center) scales on both Intrepid and Jaguar out to 32 K processes, plotted in log-log scale. Speedup of Radix-k over binary swap is shown at the right.

Bounding box and run-length encoding optimizations permitted the use of higher k-values than would otherwise be possible. The combination of accelerations with higher k-values allowed us to scale to 32,768 processes, as shown in the center panel of Figure 3. Tests were conducted at three zoom levels with the camera facing down the z-axis. The image size for these tests was 64 megapixels, a wall-size image with the same resolution as 32 HD TVs. At 32,768 processes, such an image can be composited in 0.08 seconds, or at 12.5 frames per second. The right panel of Figure 3 shows the speedup of Radix-k over binary swap in the same application, with identical optimizations applied. Up to five times faster performance resulted in some instances, and at least three times shorter compositing time was reported in most cases on Intrepid. On Jaguar, Radix-k is approximately 1.5 times faster than binary swap.

Figure 4 shows initial results of our parallel particle tracing algorithm on a dataset of the mixing behavior of warm and cold water in a confined region. The data have been resampled from their original topology onto a regular grid. Particle tracing consists of a number of iterations; one iteration involves numerically integrating the particle along the flow field until it reaches the boundary of its current block and then passing the particle to the neighboring block. Our tests ran for 10 iterations; the left panel of Figure 4 shows the result for a small number of particles, 400 in total. A single time-step of data is used here to model static flow.

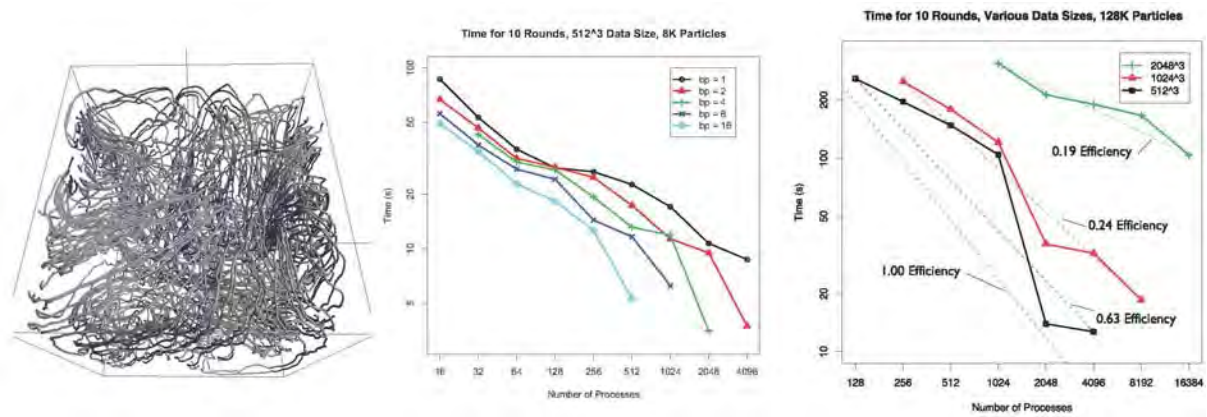


Figure 4. The scalability of parallel nearest-neighbor communication for particle tracing of thermal hydraulics data is plotted in log-log scale. The left panel shows 400 particles tracing streamlines in this flow field. The center panel shows time for 8 K particles and a data size of 512^3. Five curves are shown for different partitioning strategies: one block per process and round-robin partitioning with 2, 4, 8, and 16 blocks per process. In the right panel, 128 K particles are traced in three data sizes: 512^3 (134 million cells), 1024^3 (1 billion cells), and 2048^3 (8 billion cells). End-to-end scaling efficiency includes I/O (reading the vector dataset from storage and writing the output particle traces).

The center panel shows strong scaling up to 4,096 processes. The data size is 512^3, and this time 8,192 total particles are used. Currently a simple round-robin distribution scheme is used to balance the computational load among processes; a process contains one or more data blocks distributed in the volume. The center panel shows distributions ranging from one block per process to 16 blocks per process. In almost all cases, the performance improves as the number of blocks per process increases; load is more likely to be distributed evenly and the overhead of managing multiple blocks remains small. Obviously, there is a limit to the effectiveness of a process having many small blocks: the cost of aggregating their particles into one message increases; moreover, the ratio of their surface area to volume grows, resulting in more communicating and less computing.

We recognize that round-robin distribution does not always produce acceptable load balancing. In our case, randomly seeding a dense set of particles throughout the domain works in our favor, but this need not be the case, as demonstrated by Pugmire et al. [6]. We are investigating load-balanced partitioning under less ideal conditions.

Continuing with this example, the right panel shows the scalability of larger data size and number of particles. Here, 131,072 seeds are randomly placed in the domain. Three sizes of the same thermal hydraulics data are tested: 512^3, 1024^3, and 2048^3; the larger sizes were generated by upsampling the original size. As in the center panel, all of the data points represent end-to-end time including I/O. Tracing 131,072 particles in the smallest data size can be accomplished in 13 seconds; the medium data size in 19 seconds, and the largest data size in about 1-1/2 minutes (approximately 50% of this time is I/O). End-to-end scaling efficiency is also plotted for the three curves; in order from smallest to largest data size, these efficiencies are 63%, 24%, and 19%, respectively.

4. Summary

Data movement is a critical part of analysis operations at scale. Any nontrivial parallel decomposition of analysis tasks requires communication among processes that can amount to a significant portion of the total analysis run time. We investigated and optimized communication motifs for two commonly used operations: global reduction and nearest-neighbor communication.

Image composition in data-parallel rendering is the motivation for exploring new global reduction algorithms. With Radix-k, we reduced compositing time by up to a factor of five compared to binary swap and composed wall-size images at nearly interactive rates. We are currently implementing the Radix-k algorithm in the IceT compositing library [8], which will enable its use in production visualization tools.

Parallel particle tracing motivated our work in nearest-neighbor communication. We successfully scaled our parallel particle tracing code to 16,384 processes on data sizes up to 8 billion grid points, or 96 GB. We are actively researching load-balancing and data-partitioning algorithms, benchmarking time-varying results, and testing our code on adaptive mesh grids as well.

Acknowledgments

We thank John Blondin, Tony Mezzacappa, Paul Fischer, and Aleks Obabko, the Argonne Leadership Computing Facility, and the National Center for Computational Sciences at Oak Ridge National Laboratory. This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357. Work is also supported by DOE with agreement No. DE-FC02-06ER25777.

References

[1] Dongarra J. 2009. International exascale software project roadmap, draft report, Tech. rep. www.exascale.org.

[2] Hsu W M. 1993. *Proc. 1993 Parallel Rendering Symposium*, Segmented Ray Casting for Data Parallel Volume Rendering (San Jose, CA) pp. 7–14.

[3] Ma K L, Painter J S, Hansen C D, and Krogh M F. 1994. *IEEE Computer Graphics and Applications*, Parallel Volume Rendering Using Binary-Swap Compositing **14** pp. 59–68.

[4] Peterka T, Goodell D, Ross R, Shen H W, and Thakur R. 2009. *SC'09: Proceedings of the Conference on High Performance Computing Networking, Storage, and Analysis*, A Configurable Algorithm for Parallel Image-Compositing Applications (New York, NY, USA: ACM) pp. 1–10.

[5] Kendall W, Peterka T, Huang J, Shen H W, and Ross R. 2010. *Proceedings of Eurographics Symposium on Parallel Graphics and Visualization EG PGV'10*, Accelerating and Benchmarking Radix-k Image Compositing at Large Scale (Norrkoping, Sweden).

[6] Pugmire D, Childs H, Garth C, Ahern S, and Weber G H. 2009. *SC'09: Proceedings of the Conference on High Performance Computing Networking, Storage, and Analysis*, Scalable Computation of Streamlines on Very Large Datasets (New York, NY, USA: ACM) pp. 1–12.

[7] Peterka T, Yu H, Ross R, Ma K L, and Latham R. 2009. *ICPP 09: Proceedings of the 2009 International Conference on Parallel Processing*, End-to-End Study of Parallel Volume Rendering on the IBM Blue Gene/P (Washington, DC, USA: IEEE) pp. 566–573.

[8] Moreland K, Wylie B, and Pavlakos C. 2001. *PVG'01: Proceedings of the IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics*, Sort-last Parallel Rendering for Viewing Extremely Large Data Sets on Tile Displays (Piscataway, NJ, USA: IEEE Press) pp. 85–9.2

Expressive feature characterization for ultrascale data visualization

W Kendall,[1] M Glatter,[1] J Huang,[1] T Peterka,[2] R Latham,[2] and R B Ross[2]

[1]Department of Electrical Engineering and Computer Science, The University of Tennessee at Knoxville, Knoxville, TN 37996, USA
[2]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA

E-mail: kendall@eecs.utk.edu

Abstract. The grand challenge for visualization is to cast information into insightful content so scientists can test hypotheses and find phenomena not possible otherwise. This challenge is faced with a critical gap in the scientists' abilities to succinctly characterize phenomena of interest in their datasets. Furthermore, as applications generate larger and more complex datasets, efficiently retrieving the features of interest becomes a significant problem. In this paper we discuss recent achievements and new capabilities in the characterization and extraction of qualitative features. We also outline the necessary components needed by a backend system to use this new capability and operate in the same environment as the scientific applications. Our methods have allowed us to scale to high process counts on leadership computing resources and have also allowed us to keep pace with the growing size of scientific datasets. We discuss our efforts of examining qualitative events in cutting-edge climate data.

1. Introduction

Visualization is the primary tool for scientists among a wide array of domains to gain insight into complex phenomena, whether it is the core collapse of a supernova or major ocean eddies and currents. One of the primary components that is tied in directly with the visualization process is the ability to describe these types of features in a way that is understandable to the human and the computer. Furthermore, the retrieval of the features plays a crucial role in the interactivity of visual analysis. With dataset sizes increasing to hundreds of terabytes and beyond, feature retrieval becomes a first-class problem along with characterization.

Our recent research on feature specification methods has led to new capabilities to concisely define and visualize qualitative user interests, for example, the "beginning of spring"event in climate modeling or the extinction regions through time in fuel combustion. In one novel approach, we used a regular expression language for finding temporal trends in datasets [3]. In another, we used statistical distributions of variables in neighborhoods for visualizing local characteristics of volumetric datasets [5]. These methods, along with many other works, have one main and simple generic requirement—the issuing of Boolean range queries. This simple requirement allows for the building of powerful systems on top of the methods.

Besides new functionalities for expressive feature characterization, we also focused on scalability in anticipation of extreme-scale challenges. Our prototype system, outlined in Figure 1, successfully extracted climatic events from over a terabyte of observational satellite data in less than one minute

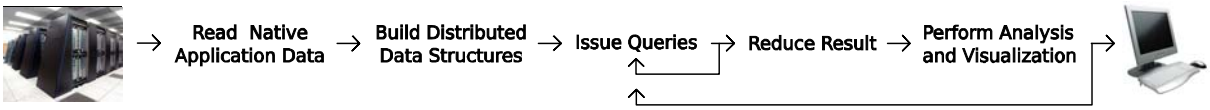


Figure 1. An overview of our end-to-end system. Data stored from a simulation is first read using advanced I/O techniques. The necessary data structures are then built for load-balanced querying to take place. After this, the user can issue queries using a higher-level language and aggregate the results. A reduction can be applied to the results before analysis and visualization.

using 16 K processes; this time included parallel I/O, parallel on-the-fly processing, and parallel visualization [6]. Harnessing this type of computing power offers a virtually unlimited ability to search for and extract features of interest. Scalability such as this is also needed for visualization to keep pace with peta- and future exascale supercomputers. In the following, we provide a holistic overview of the various components that are required by such a system, details of our methods for expressive feature characterization, and sample results from our techniques when applied to terascale observational climate data.

2. System Components

To be successful on the user and application level, the design of our end-to-end system is driven with usability and scalability in mind. We also keep the issuing of range queries as a black box for encapsulation by other higher-level tools. Figure 1 overviews the system in terms of three major components: parallel I/O, distributed data structures, and analysis.

2.1. Parallel I/O

The I/O component is the most challenging part of the system. The scalability of I/O systems is hampered by many factors, and even seemingly advanced I/O techniques might not be effective for certain problems. For usability, the system requires handling data formats familiar to the scientist, which makes the problem even more difficult.

Working with data directly from simulations, i.e., “native application” form, requires performing I/O on higher-level file formats like netCDF and HDF stored across multiple time-varying files. Our system elegantly handles data in this format by first assigning files to processes in a way to maximize contiguous I/O requests and then using the Block I/O Layer (BIL) to read data across multiple files and variables. Significant I/O bandwidth increases can be obtained when using multicollective I/O techniques [7] for reading multiple files and variables at once, especially when smaller files would not individually warrant the use of high parallelism.

This method allowed us to achieve up to 28 GB/s bandwidth performance on the Cray XT4 Jaguar machine at Oak Ridge National Laboratory, which is 75% of reported IOR benchmark performance tests [2]. Performance such as this in an application setting is critical in reducing the latency between the end of simulations and analysis and visualization.

2.2. Distributed Data Structures

In our studies [4], we have found that using a structure similar to a B-tree for query processing adheres closest to our system design goals. The ability to search arbitrary amounts of data with unlimited cardinality provides the required usability, and the potential to easily partition the tree in an embarrassingly parallel fashion supplies the needed scalability. The storage overhead of our tree structure is also minimal and usually less than one percent of the dataset size.

Beneath the tree, the data is stored at the granularity of individual items that contain the following queryable elements: scalar or vector values of the voxel they represent, spatiotemporal quantities, and other on-the-fly derived properties. The items are distributed to each process in such a way to enhance load-balanced query results; we found that randomly distributing items achieved a low distribution overhead while giving ideal load balance. This method has allowed us to achieve significant performance increases at every scale up to 16 K processes [6]. With such scalability, the querying

component of our system is effectively unbounded and allows for more sophisticated analysis and querying languages to be built on top of it.

2.3. Analysis

The analysis component of the system is built on top of a familiar parallel programming paradigm known as MapReduce [1]. In MapReduce, the user maps a value to a key and then the keys are sorted so that a reduction can be applied to the values from each key in parallel. This programming concept is applicable to a wide variety of problems in data mining and document indexing, and it can also be applied to many types of scientific analyses.

When querying has been completed in our system, the relevant items can then be sorted by a key, such as the spatial or temporal dimensions or variable values from the query. This allows for more sophisticated types of analyses such as temporal differencing or clustering to be applied. The usability of the system is further extended to more powerful querying languages that can take advantage of ordering of results; one such example from climate science that requires this feature, drought detection, and other examples are outlined in the following section.

3. Feature Characterization and Visualization

Our feature characterization is driven by the fact that the scientists' definition of the features they expect to find may be precise or vague. In both of these scenarios, the complexity of the dataset plays a major role in feature exploration. When simulations produce datasets that have many variables, timesteps, and even many models, the use of a higher-level ability to quickly and comprehensively explore data is necessary. Specifically, the ability to provide "wildcards" allows for exploring a greater degree of uncertainty about the dataset.

3.1. Regular Expression Framework

A language with the qualities of regular expressions fits in this framework and provides the ability to succinctly characterize features. The language is modeled after the work of [3]. This research showed the effectiveness of using a language to discover temporal events. For example, the query $???[SNOW \leq 0.7]*T[SNOW > 0.7]?*$ detects the first large snowfall. The first three question marks indicate that it does not matter what values of snow are present during January, February, and March. $[Snow \leq 0.7]*$ represents a low snowfall for zero or more months after March, and $T[Snow > 0.7]$ returns the first month T in which the snowfall reached the 0.7 threshold. The months after this do not matter and are represented with the question mark followed by the asterisk. T values and items that matched the query are returned to visualize the event of first snowfall.

We are currently extending this language and developing components that will allow discovery of more sophisticated features. One such example, $?*[VEG < 0.5 \ \& \ WATER < 0.3]\hat{}4?*$, is a query for drought. The caret followed by the four indicates that low water and vegetation must occur for at least four timesteps in a row to be considered a drought. We are also exploring another extension to the language that will permit querying of neighborhood and cluster-based features. Describing features in this way showed a wide variety of uses in [5] and it would be useful to interactively perform these queries on large datasets.

3.2. Case Study—MODIS

We applied our backend system to over a terabyte of observational Moderate Resolution Imaging Spectroradiometer (MODIS) data. The dataset contains vegetation and water indices over a 500-meter resolution sampling of North and South America (31,200 × 21,600 grid) spanning 417 timesteps over 9 years; each timestep is stored in a separate file. We used the Jaguar XT4 machine at Oak Ridge National Laboratory which consists of 7,832 quad-core 2.1 GHz AMD Opteron processors and a Lustre parallel file system.

To obtain a better understanding of the length of a snow season in MODIS, we issued the query represented by $?*T[0.7 \leq WATER \leq 0.9]?*T[0.4 \leq VEG \leq 0.6]?*$. The returned T variables are

the first occurrences of high water content (a likely indicator of snow) and vegetation green up; the difference of these provides an approximated length of the snow season. A visualization from the year 2006 is shown in Figure 2. Some obvious structures can be seen, such as an abnormally long snow season in the ski resorts of Colorado and the Northern Rocky Mountains in Canada. We can also observe distinct contours in snow season length when entering the Boreal Forest in Canada. Some anomalies like the Amazon Rainforest are returned because of the abnormally high water content in the area, our original definition for snow.

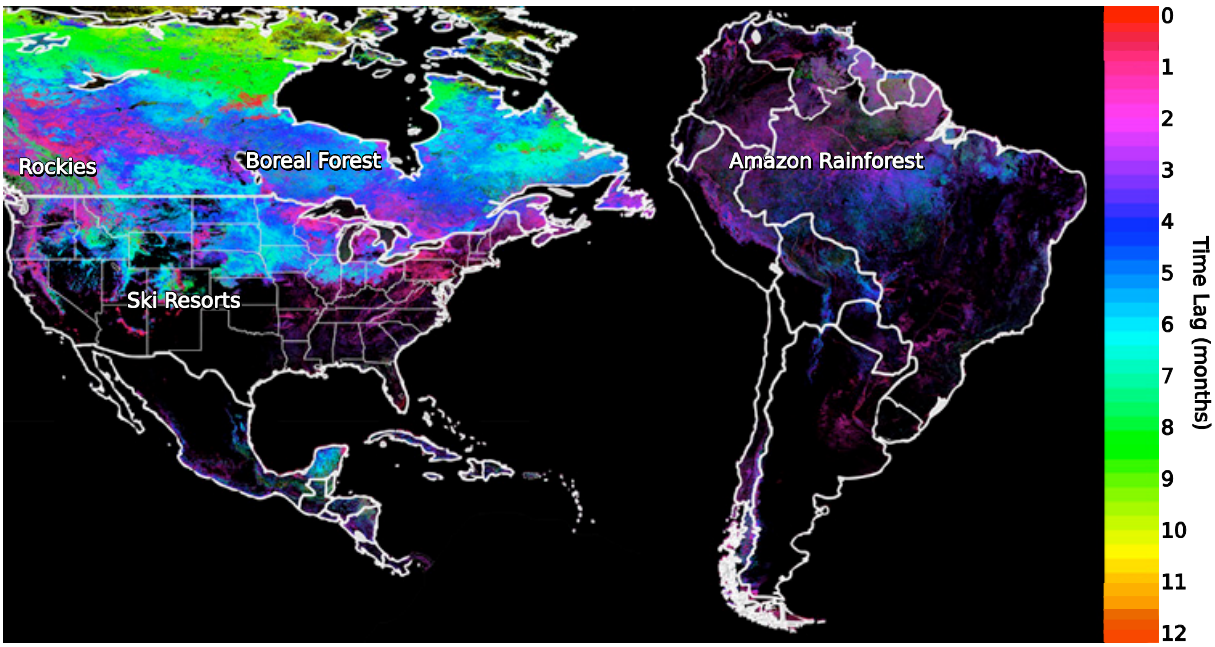


Figure. 2. Overview of the length of snow season. This visualization was created by querying the time lag between the first occurrences of abnormally high water content (an indicator for snow) and vegetation green up. Some notable areas are marked such as the ski resorts in Colorado, which have long snow seasons. Other areas like the Amazon Rainforest are returned because of the naturally high amount of water content, our original indicator for snow.

Application timing is shown in Figure 3. The reading time includes the entire time it took to query the netCDF data for variable information, assign I/O to processes, and then read the data in memory. The processing time includes filtering out useless content (e.g., ocean values), redistributing the data for load-balanced querying, and locally sorting the data. Querying, reducing, and writing the images are summarized and shown as an aggregate.

The application executed in slightly over one minute at the largest scale, a seemingly insignificant amount of time in comparison to the amount of data being processed. I/O was the dominant component of the application, taking 46 seconds at 16 K processes. We reached our peak reading rates at 2 K processes, and I/O performance degraded slightly when scaling higher; this phenomena has also been observed in other scenarios when using high process counts for I/O on Jaguar [2]. The other components of the system scaled well at high process counts. Processing the entire dataset took as little as 8 seconds. All 18 queries (2 hemispheres for each of the 9 years) were completed in 0.51 seconds and returned 11 billion relevant items. At 16 K processes, all items were reduced in 4.8 seconds for temporal analysis and it took 2 seconds overall to write the 9 images for each year.

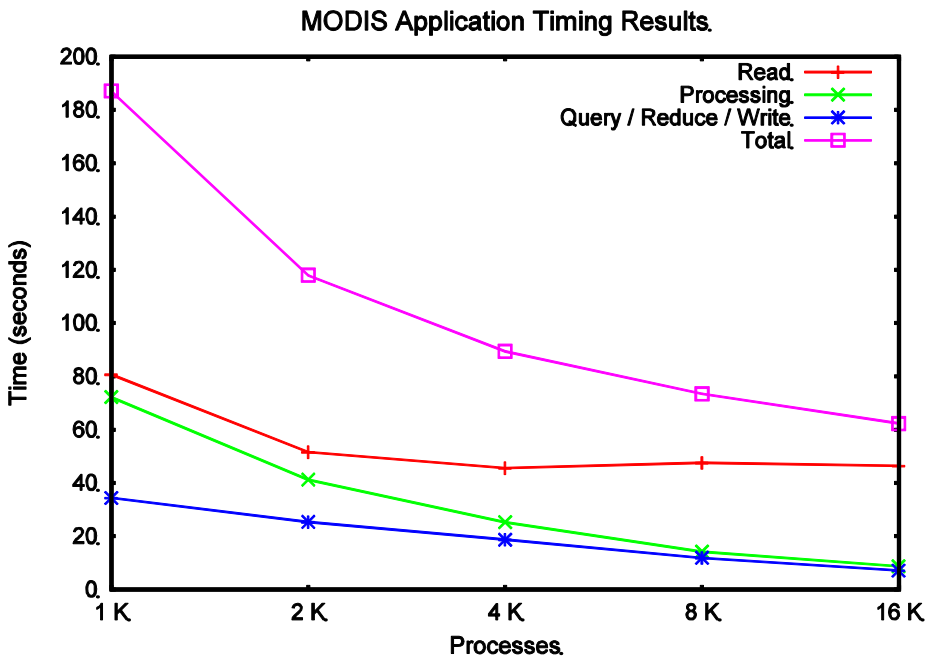


Figure 3. Timing results for the MODIS application. The results include component times for reading and processing data, along with an aggregate timing for querying, reducing, and writing results. The total application time is shown as a sum of these components.

4. Summary

The ability to describe complex events using a mini programming language is powerful. High-level languages enable intuition-led discovery of conceptual features that may involve a multitude of uncertainties. Supported by a scalable backend, our approach provides a powerful means of application science research, and also a promise of continued scalability at extreme scales.

We learned crucial lessons in the designing of an ultrascale visualization system that closely integrates sophisticated I/O together with full-range processing and analysis of data in native application formats. We have embedded our know-how into two open source packages: SQI—our backend querying system, and BIL—our multi-file parallel I/O library for visualization and analysis. Downloads are freely available at ifundefinedselectfont http://seelab.eecs.utk.edu.

Acknowledgments

Funding for this work is primarily through the Institute of Ultra-Scale Visualization (http://www.ultravis.org) under the auspices of the SciDAC program within the U.S. Department of Energy (DOE). System development was supported in part by a DOE Early Career PI grant awarded to Jian Huang (No. DE-FG02-04ER25610) and by NSF grants CNS-0437508 and ACI-0329323. MODIS data was provided by NASA (http://modis.gsfc.nasa.gov). Resources were used from the National Center for Computational Science at Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC, for DOE under Contract No. DE-AC05-00OR22725.

References

[1] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *OSDI '04: 6th Symp. on Operating System Design and Implementation*, 2004.

[2] M. R. Fahey, J. M. Larkin, and J. Adams. I/O performance on a massively parallel Cray XT3/XT4. In *IEEE Intl Symp. on Parallel and Distributed Processing*, pages 1–12, 2008.

[3] M. Glatter, J. Huang, S. Ahern, J. Daniel, and A. Lu. Visualizing temporal patterns in large multivariate data using textual pattern matching. *IEEE Trans. on Visualization & Computer Graphics*, 14(6):1467–1474, 2008.

[4] M. Glatter, C. Mollenhour, J. Huang, and J. Gao. Scalable data servers for large multivariate volume visualization. *IEEE Trans. on Visualization & Computer Graphics*, 12(5):1291–1299, 2006.

[5] C. R. Johnson and J. Huang. Distribution driven visualization of volume data. *IEEE Trans. on Visualization & Computer Graphics*, 15(5):734–746, 2009.

[6] W. Kendall, M. Glatter, J. Huang, T. Peterka, R. Latham, and R. Ross. Terascale data organization for discovering multivariate climatic trends. In *SC '09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pages 1–12, 2009.

[7] G. Memik, M. T. Kandemir, W.-K. Liao, and A. Choudhary. Multicollective I/O: A technique for exploiting inter-file access patterns. *ACM Trans. on Storage*, 2(3):349–369, 2006.

Cleaning up the cold war: simulating uranium migration at the Hanford 300 Area

Glenn Hammond,[1] Peter Lichtner[2]

[1]Hydrology Group, Energy and Environment Directorate, Pacific Northwest National Laboratory, Richland, WA, 99352
[2]Hydrology, Computational Earth Sciences Group, Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, 87545

E-mail: glenn.hammond@pnl.gov and lichtner@lanl.gov

Abstract. The reactive multiphase flow and geochemical transport code PFLOTRAN, developed under the SciDAC-2 program, is employed to simulate groundwater flow and uranium [U(VI)] contaminant transport at the Hanford 300 Area. Simulations predict that ~25 kg/yr of U(VI) leach into the neighboring Columbia River. Simulations further predict that heterogeneous permeability, modeled through stochastically-generated random fields, has little impact on the flux of U(VI) to the river.

1. Introduction

Throughout the Cold War, facilities at the Hanford site in south central Washington State were built and utilized to produce plutonium in support of nuclear weapons production. During this period of time, many regions of the Hanford site were contaminated with waste byproducts resulting from the nuclear weapons production process. Cleanup operations at the Hanford site have been ongoing since the close of the Cold War in the mid-1980's, but many of the contaminants at the site still exist in the subsurface. In particular, at the Hanford 300 Area, where nuclear fuel rods were fabricated, uranium [U(VI)] has persisted in the underlying aquifer far longer than the original estimates for natural attenuation [1,2]. These erroneous estimates resulted from uncertainty regarding the extent of the source zone(s), which remains uncertain, and the use of an overly simplistic process model for U(VI) sorption (i.e., the KD model) [3]. In order to better understand the persistence of U(VI) at the Hanford 300 Area and its leach rate into the neighboring Columbia River, variably-saturated groundwater flow and multicomponent geochemical transport simulations are being employed to simulate U(VI) migration at the site using the massively-parallel, reactive multiphase flow and geochemical transport code PFLOTRAN [4]. PFLOTRAN is currently being developed under the Department of Energy Office of Science SciDAC-2 program to advance scientific understanding of subsurface processes and improve predictive flow and transport capability. In these PFLOTRAN simulations, high-performance computing is employed to preserve numerical accuracy through high-resolution computational grids and accommodate the large number of chemical species and small time steps needed to accurately depict river-groundwater interaction.

In the sections that follow, the Hanford 300 Area is described in further detail, along with the governing equations employed within PFLOTRAN and the Hanford 300 Area conceptual model. Estimates of the U(VI) flux to the Columbia River are then presented for several conceptual models of soil permeability and U(IV) sorption at the site.

2. Background

The Hanford 300 Area is located in the southeast corner of the Hanford site along the western bank of the Columbia River, just north of the city of Richland, Washington as shown in Figure 1. Facilities at the 300 Area were used to fabricate nuclear fuel rods for plutonium production during the Cold War. During fuel rod production, U(VI) was discharged into several process ponds (i.e., the North and South Process Ponds) and nearby trenches. This U(VI) infiltrated into the underlying soils and aquifer contaminating the groundwater in the region. Over time, this contaminated groundwater has migrated towards the river, slowly seeping into the river through the hyporheic zone at the river bottom.

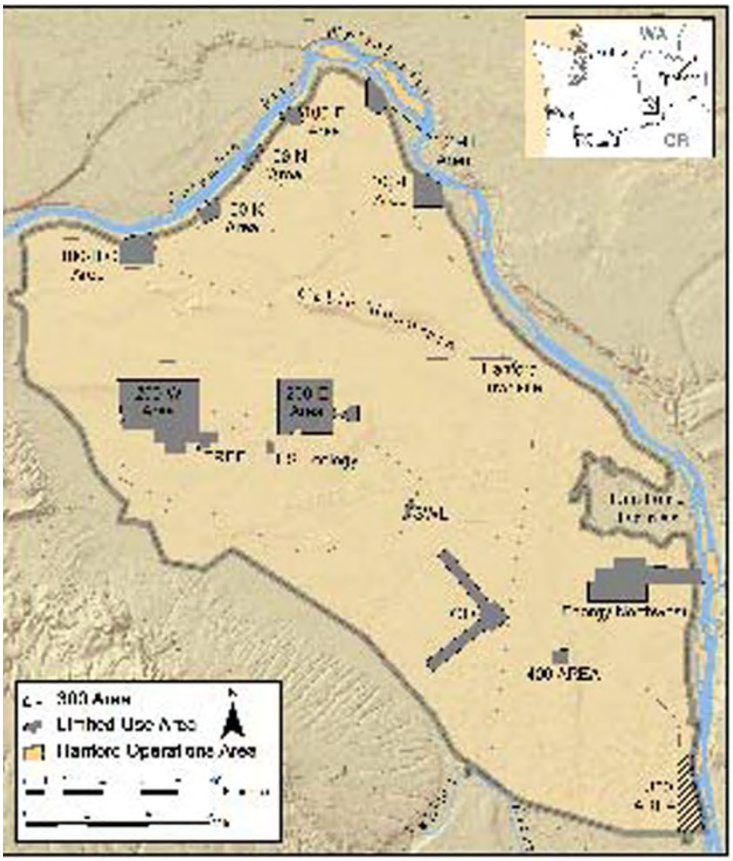


Figure 1. Map of the Hanford Site showing the location of the Hanford 300 Area to the lower right.

The undulating Columbia River stage plays a dominant role in controlling groundwater flow within the aquifer underlying Hanford 300 Area. River stage fluctuates ~2.5 meters annually due to seasonal variation with daily fluctuations of up to ~1.5 meters due to cyclic spillage from the Priest Rapids dam upriver. Due to the high permeability of aquifer materials, water flows readily in and out of Hanford 300 Area aquifer when river stage is high and low, respectively. Thus, the river stage drives cyclic groundwater flow and consequently the migration of the U(VI) plume at the site. However, sorption of U(VI) to soil grains is so strong that the contaminant travels much slower than the groundwater pore velocities.

The real challenge posed by the U(VI) plume at the 300 Area is its persistence due to sorption onto aquifer materials. Regulators desire that groundwater concentrations be reduced from current levels of over 100 μg/L to below the maximum contaminant limit (MCL) of 30 μg/L for U(VI). Although this reduction in concentration will surely occur as a result of natural attenuation of U(VI) as it slowly leaches in to the river, the time frame of natural attenuation is uncertain; the plume could persist for

decades to centuries. Assuming that the 300 Area will be remain a brown field (i.e., off-limits to the public) for years to come, the question then becomes what is the flux of U(VI) to the river over the course of a year and is this U(VI) flux acceptable? Although the latter question is for regulators to decide, quantification of the U(VI) flux to the river is possible through simulation of groundwater flow and geochemical transport using PFLOTRAN.

3. Governing Equations

The conceptual model employed in this work for simulating U(VI) migration at the Hanford 300 Area incorporates groundwater flow in both the vadose and saturated zones and multicomponent geochemical transport (i.e., aqueous speciation, surface complexation, and mineral precipitation-dissolution). The PFLOTRAN governing equations for single-phase variably saturated groundwater flow and geochemical transport in porous media are detailed below.

3.1. Variably-Saturated Flow

PFLOTRAN's variably-saturated flow equations based on the Richards equation [5] have the form

$$\frac{\partial}{\partial t}(\varphi s \rho) + \boldsymbol{\nabla} \cdot \rho \boldsymbol{q} \;=\; 0, \tag{1}$$

with water density ρ, porosity φ, water saturation s, and Darcy velocity $\boldsymbol{q}$ given by

$$\boldsymbol{q} \;=\; -\frac{k k_r}{\mu} \boldsymbol{\nabla} (p - \rho g z), \tag{2}$$

with pressure p, viscosity μ, acceleration of gravity g, intrinsic permeability k, relative permeability k_r, and z the component of the distance vector aligned with gravity. The van Genuchten relation [6] is used to relate capillary pressure to water saturation and the Burdine correlation [7] is used for relative permeability.

3.2. Geochemical Transport

PFLOTRAN's multicomponent reactive transport equations have the form

$$\frac{\partial}{\partial t}\left(\varphi s \Psi_j + \Psi_j^S\right) + \boldsymbol{\nabla} \cdot \boldsymbol{\Omega}_j \;=\; -\sum_m \nu_{jm}^{\min} I_m. \tag{3}$$

In this equation the quantity Ψ_j denotes the total aqueous concentration of the jth primary species

$$\Psi_j \;=\; C_j + \sum_i \nu_{ji}^{\mathrm{aq}} C_i, \tag{4}$$

where the concentration of the ith secondary species C_i is expressed as a function of primary aqueous species activities through the mass action equation

$$C_i \;=\; \frac{K_i}{\gamma_i} \prod_j \left(\gamma_j C_j\right)^{\nu_{ji}^{\mathrm{aq}}}, \tag{5}$$

with equilibrium constant K_i, stoichiometry ν_{ji}^{aq} and activity coefficients $\gamma_{i,j}$

The quantity Ψ_j^S denotes the total sorbed concentration of the jth primary species

$$\Psi_j^S \;=\; \sum_{k\sigma} \nu_{jk}^{\mathrm{srf}} S_{k\sigma}, \tag{6}$$

where $S_{k\sigma}$ denotes the surface complex concentration of the kth species for site σ. PFLOTRAN supports both equilibrium and multirate kinetic formulations of surface complexation. However, for brevity only the equilibrium formulation is described in this work. The multirate formulation is

described in detail by [8]. For equilibrium surface complexation, concentrations of surface complexes $S_{k\sigma}$ and empty surface sites $S_{\chi\sigma}$ satisfy the site conservation equation of the form

$$\omega_\sigma = S_{\chi\sigma} + \sum_k S_{k\sigma}, \tag{7}$$

for sorption site σ. The site concentration ω_σ [mol sites/m^3] is related to the specific site surface density η_σ [mol sites/m^2] of the medium by the relation

$$\omega_\sigma = \eta_\sigma A_\sigma \rho_r (1-\varphi). \tag{8}$$

In this equation the quantity η_σ denotes the moles of surface sites of type σ per unit area, A_σ represents the specific sorptive surface area in units of m^2/g, and ρ_r refers to the rock grain density. The concentrations for unoccupied sites $S_{\chi\sigma}$ and surface complex $S_{k\sigma}$ are found to be

$$S_{\chi\sigma} = \frac{\omega_\sigma}{1+\sum_k K_k Q_k}, \tag{9}$$

and

$$S_{k\sigma} = \frac{\omega_\sigma K_k Q_k}{1+\sum_{k'} K_{k'} Q_{k'}}, \tag{10}$$

where the ion activity product Q_k is given by

$$Q_k = \prod_j (\gamma_j C_j)^{\nu_{jk}^{\mathrm{srf}}}. \tag{11}$$

The total solute flux Ω_j is defined by

$$\mathbf{\Omega}_j = \left(\boldsymbol{q} - \varphi s \tau D \boldsymbol{\nabla}\right)\Psi_j, \tag{12}$$

with tortuosity τ and diffusion/dispersion coefficient D. The form of the flux in Equation (12) assumes diffusion coefficients to be species independent.

Mineral reaction rates I_m have the form based on transition state theory given by

$$I_m = -k_m a_m \mathcal{P}_m \left(1 - K_m Q_m\right) \zeta_m, \tag{13}$$

where for the mth mineral, k_m denotes the kinetic rate constant, a_m represents the specific surface area, P_m denotes a prefactor to account for the dependence of the rate on pH and other species, K_m denotes the equilibrium constant, and Q_m the ion activity product defined by

$$Q_m = \prod_j (\gamma_j C_j)^{\nu_{jm}^{\mathrm{min}}}. \tag{14}$$

The factor ζ_m takes on the values zero or one to ensure that if a mineral is not present at some m particular point in space, it cannot dissolve:

$$\zeta_m = \begin{cases} 1, & \varphi_m > 0 \text{ or } K_m Q_m > 1, \\ 0, & \varphi_m = 0 \text{ and } K_m Q_m \le 1 \end{cases}. \tag{15}$$

The mineral mass transfer equation has the simple form

$$\frac{\partial \varphi_m}{\partial t} = \overline{V}_m I_m, \tag{16}$$

where $\bar{V}_m$ and φ_m refer to the molar volume and volume fraction, respectively, of the *m*th mineral.

4. Hanford 300 Area Conceptual Model

4.1. Spatial Discretization

The Hanford 300 Area model considers a three-dimensional problem domain measuring 900 × 1300 × 20 meters (*x*, *y*, *z*) in size, rotated at 14° counter-clockwise about the *z*-axis at the origin for alignment with the Columbia River as illustrated with the yellow box in Figure 2. The base of the model domain lies at 90 meters elevation above sea level. The discretized grid is composed of 1,872,000 uniformly-spaced grid cells with 5-meter horizontal and 0.5-meter vertical grid spacing.

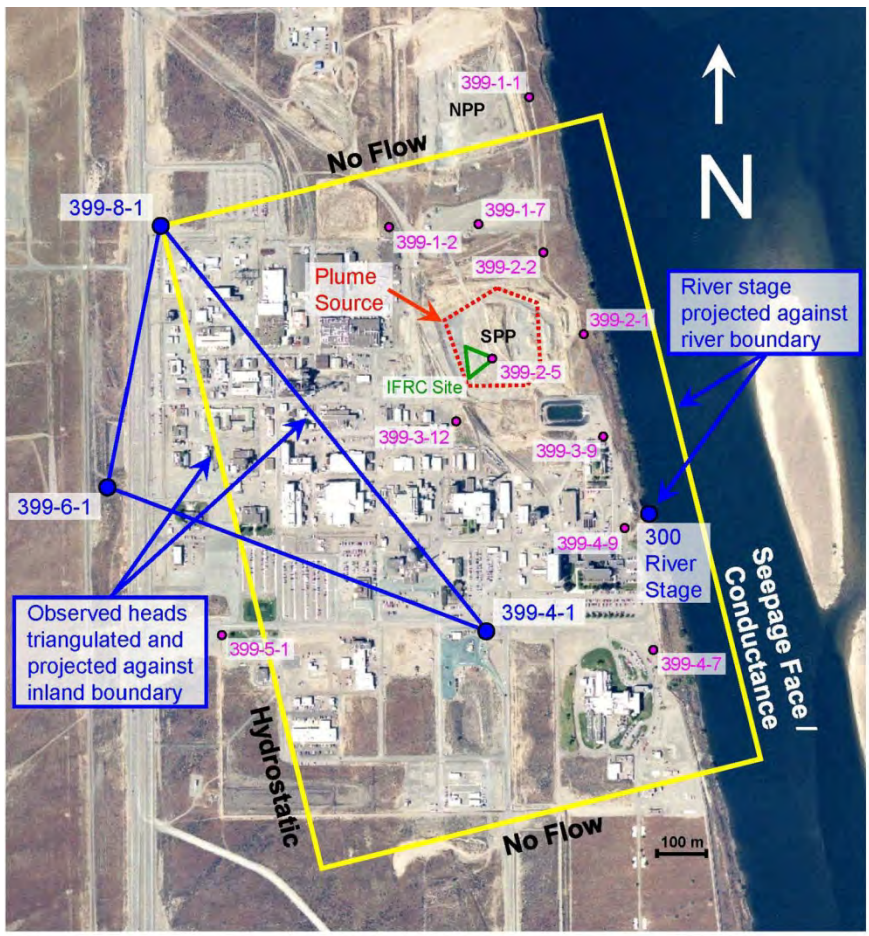


Figure 2. Planar view of outline of 3D computational domain (yellow line) showing the locations of the North and South Process Ponds (NPP & SPP), the IFRC field site, and a subset of observation wells. The transient inland boundary condition at the western edge is constructed by projecting head data obtained from wells 399-8-1, 399-6-1 and 399-4-1 onto the western boundary. The transient Columbia River stage is projected onto the river boundary to the east. No flow boundary conditions are imposed on the north and south faces.

4.2. Hydrogeology

The geologic conceptual model for the Hanford 300 Area is based on a three-dimensional solids model of the site constructed by delineating the ground surface, river bathymetry, and underlying hydrostratigraphic units. For this modeling effort, three geologic units are considered: (1) Hanford sands and gravels, (2) Ringold fine-grained silts, and (3) Ringold gravels. The Hanford unit extends from the surface of the site through the vadose zone to below the water table with the contact between the Hanford and Ringold units forming an undulating surface, as viewed from the southeast in Figure 3. Hydraulic properties for these units (i.e., porosity φ, tortuosity τ, van Genuchten parameters α and m, residual saturation S_r, and permeability k) were obtained from [9]. The permeability of the upper Hanford unit, which governs the rate at which fluid flows through it, is two to three orders of magnitude larger than that of the lower Ringold units. Therefore, most groundwater flow to the river is through the Hanford unit in the upper portion of the model domain.

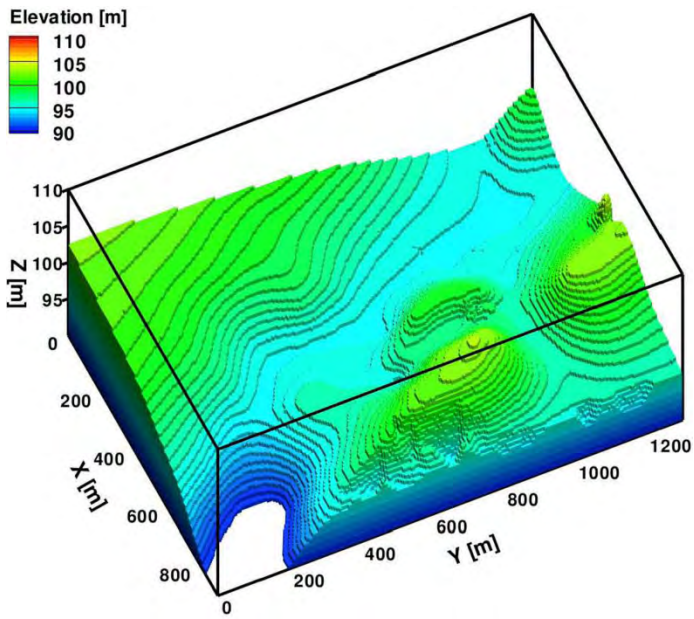


Figure 3. Hanford-Ringold contact with elevation contoured on the Ringold units and the upper Hanford unit hidden.

Material properties are specified within the computational domain by superimposing the PFLOTRAN grid on the 3D solids model and assigning material IDs to cells based on their location within the solids model. Grid cells above the ground surface and river bottom are considered inactive, while active cells are assigned material properties based on cell material IDs. Permeabilities for stochastic simulations incorporating heterogeneity are assigned only to the Hanford unit since the lower Ringold units could be considered relatively impermeable, though they clearly are not.

4.3. Geochemistry

The geochemical conceptual model consists of 15 primary aqueous species (H^+, Ca^{2+}, Cu^{2+}, Mg^{2+}, UO_2^{2+}, K^+, Na^+, HCO_3^-, Cl^-, F^-, HPO_4^{2-}, NO_3^-, SO_4^{2-} and two tracers), 88 secondary 2 3 434 aqueous complexes, two kinetically-reacting minerals (Calcite and Metatorbernite), two surface complexes (>$SOUO_2OH$ and >$SOHUO_2CO_3$) and one surface site (>SOH). Important geochemical reactions for equilibrium surface complexation and mineral dissolution include:

$$>\mathrm{SOH} + \mathrm{UO_2^{2+}} - 2\ \mathrm{H^+} + \mathrm{H_2O} \rightleftharpoons >\mathrm{SOUO_2OH}, \tag{17a}$$

$$>\mathrm{SOH} + \mathrm{UO_2^{2+}} - 2\ \mathrm{H^+} + \mathrm{H_2O} + \mathrm{CO_2} \rightleftharpoons >\mathrm{SOHUO_2CO_3}, \tag{17b}$$

$$2\ \mathrm{UO_2^{2+}} + \mathrm{Cu^{2+}} + 2\ \mathrm{HPO_4^{2-}} + 8\ \mathrm{H_2O} - 2\ \mathrm{H^+} \rightleftharpoons \mathrm{Metatorbernite(s)}, \tag{17c}$$

$$\mathrm{Ca^{2+}} + \mathrm{CO_2} - 2\ \mathrm{H^+} + \mathrm{H_2O} \rightleftharpoons \mathrm{Calcite(s)}. \tag{17d}$$

Selectivity coefficients for the two surface complexes are taken from [10]. Mobility of uranium is also a affected by aqueous complexation reactions with the two dominant species being $\mathrm{CaUO_2(CO_3)_3^{2-}}$ and $\mathrm{Ca_2UO_2(CO_3)_{3(aq)}}$. Overall, 28,080,000 geochemical degrees of freedom are represented in the model. Further details regarding the geochemistry (e.g., initial and boundary concentrations, dominant aqueous U(VI) species, etc.) may be found in [8].

4.4. Boundary and Initial Conditions

The western boundary condition for the model is a transient hydrostatic condition with pressures calculated through the triangulation and interpolation of inland well head data and projection along the western face. The eastern river boundary is a dynamic seepage face/conductance condition (above and below river stage, respectively) that mimics the influence of the river bank and river bottom sediment on river-groundwater interaction. The transient eastern boundary pressures are calculated from river stage and river gradient data. No flow boundary conditions are assigned to the north and south boundaries of the model, while recharge due to precipitation is modeled through a specified flux at the top surface.

The model initial condition is set by running a simulation without sorption for several years where the U(VI) plume emanates from the source zone in the foot print of the South Process Pond (see Figure 2) and is transported toward the river. After several years, this plume reaches a quasi steady state and does not grow significantly in size. At this point, the simulation is checkpointed, and upon restart the sorbed U(VI) concentrations are equilibrated with the aqueous phase geochemistry. The model then proceeds through the year-long simulation with sorption incorporated in the geochemistry. This “restart” approach allows the model to bypass the U(VI) plume emplacement phase in time, the which is highly uncertain with regard to the magnitude and time of release.

5. Simulation Results

As discussed in Section 2, numerical simulation can be employed in an attempt to predict the flux of U(VI) from the Hanford 300 Area to the Columbia River. PFLOTRAN simulations were run based on the conceptual model described in Section 4 to estimate this flux. Both homogeneous and heterogeneous permeability fields were considered in these simulations. For the homogeneous simulations, 4096 processor cores were employed for ~5 hours on Oak Ridge National Laboratory’s Jaguar XT4/5 supercomputer to complete a 1-year simulation. In the case of the stochastic simulations with ten heterogeneous permeability fields, PFLOTRAN’s multi-realization capability was utilized to run all ten simulations simultaneously on 40,960 processor cores, each simulation employing 4096 cores. These stochastic simulations required an increased 10–11 hours to complete (as opposed to the ~5 hours for the homogeneous runs), primarily due to an increased number of linear solver iterations in the flow solve, likely a result of incorporating heterogeneity. Altogether, the ten stochastic simulations consumed ~450K cores hours on Jaguar. Assuming a conservative 50% parallel efficiency for PFLOTRAN, these simulations would have required ~25 years to run on a single desktop computer, assuming the sufficient memory were available.

Figure 4 illustrates the cumulative U(VI) flux to the Columbia River of ~25 kg/yr for a homogeneous Hanford unit permeability field. This 25 kg/yr result is in close agreement with estimates of 20–50 kg/yr predicted by field hydrologists at the site (Peterson, pers. comm.). Figure 4 depicts river stage in the background with cumulative U(VI) flux observed for the cases of no sorption, equilibrium sorption, and multirate kinetic sorption, where sorption and surface

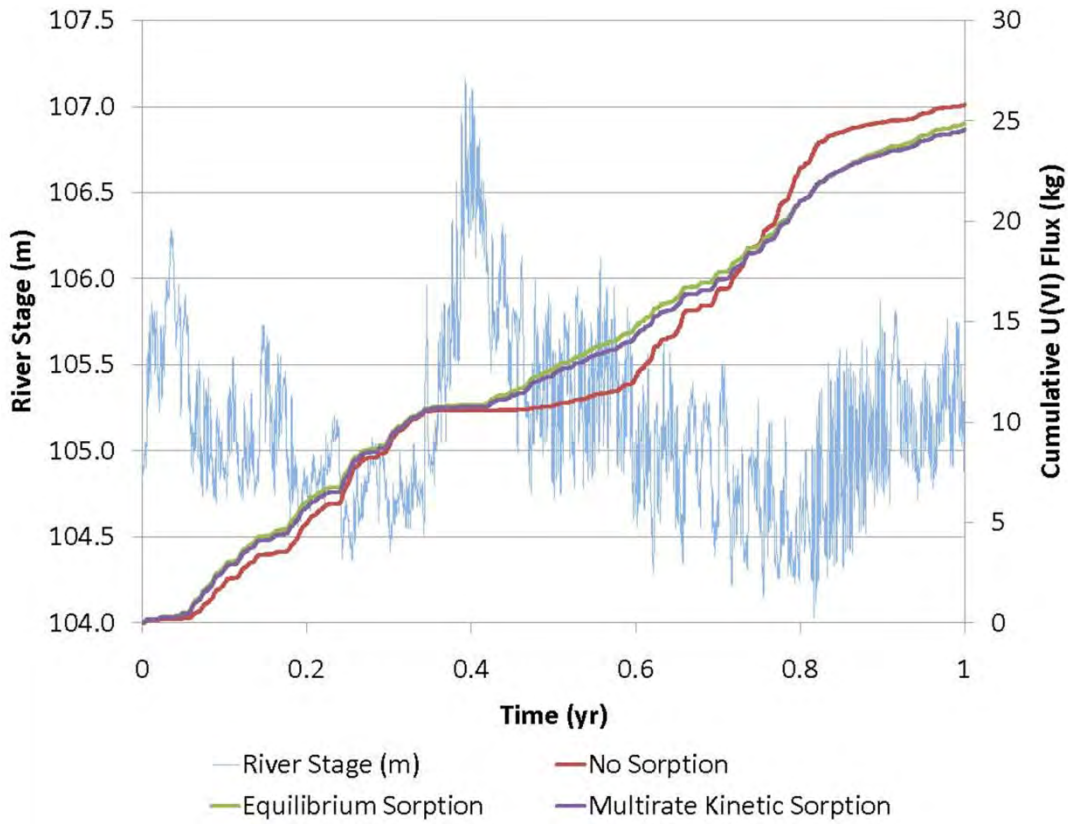


Figure 4. Cumulative flux for equilibrium and multirate sorption, and no sorption, at the river-aquifer boundary plotted as a function of time.

complexation are synonymous. From the plot it is evident that there is little difference between the equilibrium and multirate kinetic surface complexation formulations for U(VI) sorption. This result is likely true only for a conceptual model that assumes that the plume spans the distance from the source zone to the river without the source becoming depleted. One would expect the flux based on equilibrium surface complexation to be higher once the non-labile U(VI) in the source zone ceases to exist.

The cumulative U(VI) flux for the non-sorbing case is similar to that of both sorption cases. This result is non-intuitive but can be explained. For both the equilibrium and multirate kinetic surface complexation scenarios, the number of sorption sites that are "occupiable" are occupied from the beginning of the simulation (recall that the sorbed U(VI) is equilibrated with the aqueous geochemistry for an initial condition). Thus, there are few to no occupiable sites available for surface complexation as the simulation proceeds and the U(VI) is transported to the river in an essentially unretarded manner. Clearly, the non-sorbing and sorbing results are not identical, particularly when river water seeps far inland from 0.35–0.55 years time pushing the U(VI) plume to the west for the non-sorbing case. However, it is evident that both surface complexation scenarios exhibit a nearly-linear mass rate or flux to the river of similar magnitude to the non-sorbing case, which suggests the lack of occupiable sorption sites.

To illustrate the impact of heterogeneous permeability on the U(VI) flux to the river, ten stochastically-generated heterogeneous permeability fields were incorporated within the conceptual model. The parameters and procedure employed to generated these random data sets are described in detail by [11]. Figure 5 illustrates a representative realization of one of these heterogeneous fields.

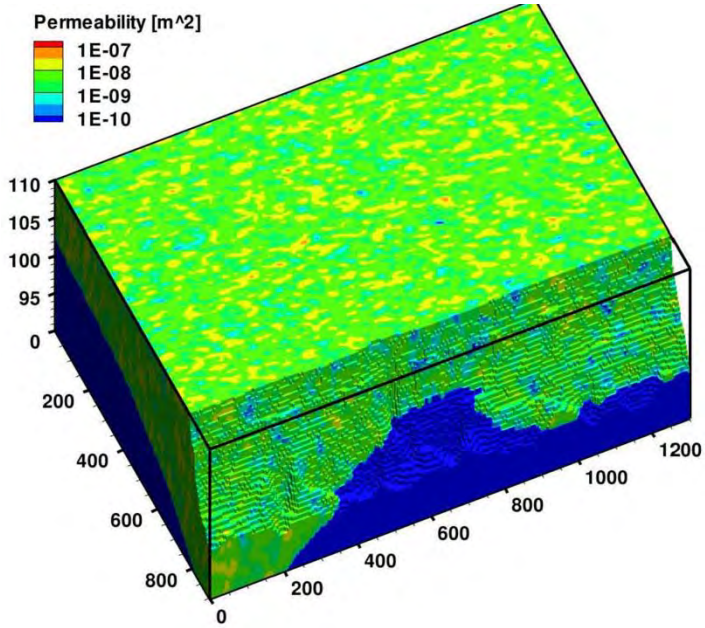


Figure 5. Heterogenous k_{xx} field for stochastic simulation. *Z*-magnification = 32×.

Figure 6 shows the variability in the cumulative U(VI) flux for the ten realizations of heterogeneous permeability, where all ten cumulative fluxes at one year are within 7% of each other. This result demonstrates a low sensitivity of the U(VI) flux to the heterogeneous permeability fields employed within the model. It should be noted that these heterogenous fields were generated based on data sets provide by the Hanford 300 Area Integrated Field Research Challenge (IFRC) project, which occupies a plot-scale region much smaller than the plume-scale conceptual model presented in this work. It is uncertain whether the plot-scale IFRC data sets can be accurately extrapolated to the plume scale. [11] provides a detailed analysis of the strengths and weaknesses of this stochastic groundwater study and discusses the potential need for incorporating larger-scale heterogeneities within the plume-scale conceptual model to better quantify uncertainty.

6. Conclusion

In this work, the reactive multiphase flow and geochemical transport code PFLOTRAN was employed to simulate migration of U(VI) at the Hanford 300 Area and quantify the flux of U(VI) leaching into the neighboring Columbia River over the course of a year. Simulations run on thousands to tens of thousands of processor cores on ORNL's Jaguar supercomputer predicted that approximately 25 kilograms of U(VI) are discharged into the Columbia River each year. The simulations also demonstrated that this U(VI) flux estimate is only slightly sensitive to small-scale heterogeneity in permeability within the site's geology. Future plans for simulation at the site include the utilization of PFLOTRAN's adaptive mesh refinement capability (currently being implemented) to improve numerical accuracy near the Hanford 300 Area IFRC site.

7. Acknowledgements

This research is supported under the U.S. Department of Energy SciDAC-2 (Scientific Discovery through Advanced Computing) program with funding provided by DOE Offices of Biological & Environmental Research (BER) and Advanced Scientific Computing Research (ASCR). Supercomputing resources were provided by the DOE Office of Science Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program with allocations on NCCS Jaguar at Oak Ridge National Laboratory.

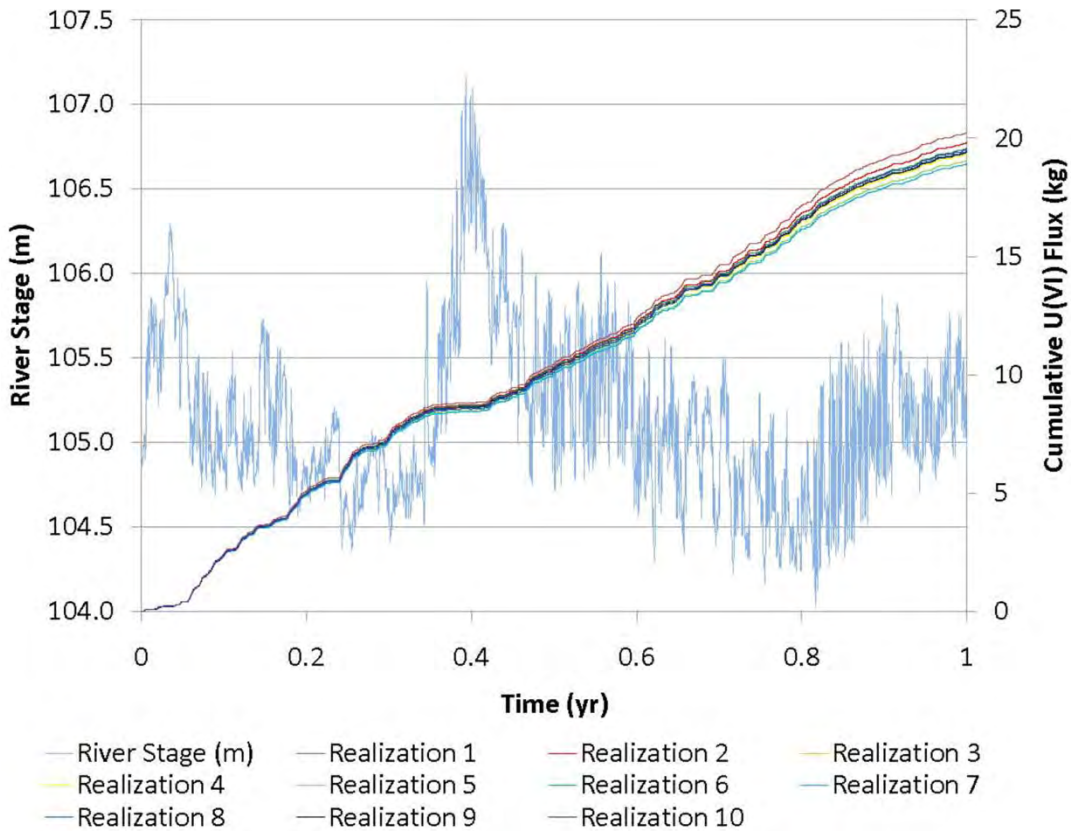


Figure 6. Comparison of cumulative U(VI) flux to Columbia River for ten realizations of permeability.

References

[1] DOE. 1994. Remedial investigation/feasibility study report for the 300-FF-5 Operable Unit, Report DOE/RL94-85, U.S. Department of Energy, Richland Operations Office, Richland, Washington.

[2] ROD. 1996. Declaration of the Record of Decision, U.S. DOE Hanford 300 Area-FF-1 and Hanford-FF-5 Operable Units, Hanford Site, Benton County, Washington Tech. rep., Washington State Department of Ecology, U.S. Environmental Protection Agency, and U.S. Department of Energy, Richland, Washington.

[3] Zachara J M, Davis J A, Liu C, McKinley J P, Qafoku N, Wellman D M, and Yabusaki S. 2005. Uranium geochemistry in vadose zone and aquifer sediments from the 300 Area uranium plume, Report PNNL-15121, Pacific Northwest National Laboratory, Richland, Washington.

[4] Hammond G E, Lichtner P C, Mills R T, and Lu C. accepted *Ground Water Reactive Transport Models*. chap PFLOTRAN: Reactive Flow & Transport Code for Use on Laptops to Leadership-Class Supercomputers.

[5] Richards L A. 1931. *Physics* **1** 318–333.

[6] van Genuchten M T. 1980. *Soil Science Society of America Journal* **44** 892–898.

[7] Burdine N. 1953. *Transactions of the American Institute of Mining and Metallurgical Engineers* **198** 71–78.

[8] Hammond G E and Lichtner P C. accepted *Water Resources Research*.

[9] Williams M D, Rockhold M L, Thorne P D, and Chen Y. 2008. Three-dimensional groundwater models of the 300 Area at the Hanford Site, Washington State Report PNNL-17708 Pacific Northwest National Laboratory, Richland, Washington.

[10] Bond D L, Davis J A, and Zachara J M. 2008. *Adsorption of Metals by Geomedia II: Variables, mechanisms, and model applications* 375–416.
[11] Hammond G E, Lichtner P C, and Rockhold M L. 2010. *Journal of Contaminant Hydrology* doi:10.1016/j.jconhyd.2010.04.005.

Evaluating variable switching and flash methods in modeling carbon sequestration in deep geologic formations using PFLOTRAN

C. Lu,[1] P.C. Lichtner,[2] G.E. Hammond,[3] R.T. Mills[4]

[1]Energy & Geosciences Institute, University of Utah, clu@egi.utah.edu
[2]Los Alamos National Laboratory, lichtner@lanl.gov
[3]Pacific Northwest National Laboratory, glenn.hammond@pnl.gov
[4]Oak Ridge National Laboratory, rmills@ornl.gov

Abstract. PFLOTRAN a massively parallel computer code for modeling coupled hydro-thermalchemical processes in variably saturated, non-isothermal porous media is applied to sequestration of supercritical CO_2 in deep geologic formations. Two different methods of solution to the governing partial differential equations are implemented referred to as variable switching and the flash approach. Variable switching entails choosing the independent variables according to the set of phases present in a control volume, whereas in the flash approach a persistent set of variables are used through the calculation. The features and performance of the two approaches are described and contrasted in regard to stability and convergence, flexibility of choice of solver, and scaling behavior.

1. Introduction

Application of high performance computing to assess the risks involved in carbon sequestration in deep geologic formations is an important issue in mitigating global warming caused by release of green house gases into the atmosphere. Models simulating multiphase subsurface reactive flows are needed which apply to basin-scale systems with three-dimensional computational domains on the order of 100 km × 100 km × 5 km. The model must account for hundreds of injection wells to accommodate the volume of CO_2 that must be injected requiring localized high spatial resolution around each well. Additional processes involving chemical reactions and mechanical effects increase the complexity and computational resources that are needed. One approach is to use adaptive mesh refinement (AMR) to refine the grid where needed in the domain, such as around wells and interfaces between different stratigraphic units. AMR is based on multilevel solvers and conventionally requires a single set of variables common to all levels. This work investigates use of the flash approach to replace variable switching methods which allows for a persistent set of unknown variables as phase changes take place.

2. Modeling Multiphase Flow

The general governing equations for a multiphase system may be written as

$$\sum_{\alpha} \left\{ \frac{\partial}{\partial t}(\varphi s_\alpha \eta_\alpha x_{i\alpha}) + \boldsymbol{\nabla} \cdot \boldsymbol{F}_{i\alpha} \right\} = Q_i, \tag{1}$$

where φ. denotes the porosity of the geologic formation, s_α, η_α refer to the saturation and molar density of phase α, $x_{i\alpha}$ denotes the mole fraction of the ith component in phase α, and Q_i denotes a source/sink term associated with the ith component. The flux $\mathbf{F}_{i\alpha}$ contains contributions from advection derived from Darcy's law and diffusion/dispersion

$$\boldsymbol{F}_{i\alpha} = \boldsymbol{q}_\alpha \eta_\alpha x_{i\alpha} + \boldsymbol{j}_{i\alpha}, \tag{2}$$

with Darcy velocity

$$\boldsymbol{q}_\alpha = -\frac{kk_\alpha}{\mu_\alpha}\boldsymbol{\nabla}\left(p_\alpha - W_\alpha \eta_\alpha g \boldsymbol{z}\right), \tag{3}$$

where formation permeability is denoted by k, relative permeability by k_α, fluid viscosity by μ_α, fluid pressure by p_α, acceleration of gravity by g, formula weight by W_α, and diffusive flux by $\mathbf{j}_{i\alpha}$ defined as

$$\boldsymbol{j}_{i\alpha} = -\varphi s_\alpha D_\alpha \eta_\alpha \boldsymbol{\nabla} x_{i\alpha}. \tag{4}$$

For an isotropic medium, the dispersion tensor is given by

$$D_\alpha = \left(D_\alpha^0 + \alpha_T \upsilon_\alpha\right)\boldsymbol{I} + \left(\alpha_L - \alpha_T\right)\frac{\boldsymbol{\upsilon}_\alpha \boldsymbol{\upsilon}_\alpha}{\upsilon_\alpha}, \tag{5}$$

with species-independent diffusion coefficient D_α^0 and fluid velocity $\upsilon_\alpha = \varphi q_\alpha$. The diffusive/ dispersive flux satisfies the condition

$$\sum_i \boldsymbol{j}_{i\alpha} = 0. \tag{6}$$

These equations are subject to the constraint conditions

$$\sum_\alpha s_\alpha = 1, \tag{7a}$$

$$\sum_i x_{i\alpha} = 1, \tag{7b}$$

the solubility constraints

$$x_{i\beta} = K_i^{\alpha\beta} x_{i\alpha}, \tag{8}$$

and capillary pressure relations

$$p_{\alpha\beta}^c = p_\alpha - p_\beta. \tag{9}$$

Constitutive relations are needed to relate capillary pressure and phase saturation such as van Genuchten or Brooks-Corey relations.

There are $N_C + 1$ unknowns and an equal number of equations, where N_C refers to the number of independent components in the system in addition to pressure and temperature. Summing Eqn. (1) over all species the diffusive flux disappears yielding the flow equation

$$\sum_\alpha \left\{\frac{\partial}{\partial t}\left(\varphi s_\alpha \eta_\alpha\right) + \boldsymbol{\nabla}\cdot\left(\boldsymbol{q}_\alpha \eta_\alpha\right)\right\} = \sum_i Q_i. \tag{10}$$

This equation may be substituted for one of the equations in Eqn. (1) such as H_2O, for example. The mass conservation equations are coupled to the energy conservation equation providing an equal number of equations as unknowns.

2.1. Methods of Solution

2.1.1. Variable Switching. In the variable switching approach the independent variables are chosen based on the phases present. This choice is not unique. Several different possibilities are listed in Table 1. An important drawback of the variable switching approach is that it may be difficult to apply when using multilevel solvers where the independent variables are different at different levels [1].

Table 1. Possible choices of independent variables used in the variable switching approach. Note that g = supercritical CO_2 = SC, and *l* = H_2O phase.

Phase	Variables								
	X_1	X_2	X_3	X_1	X_2	X_3	X_1	X_2	X_3
Liquid	P_l	T	$X^l_{CO_2}$	p_l	T	$X^l_{CO_2}$	p_l	T	$X^l_{CO_2}$
Gas	p_g	T	$X^g_{CO_2}$	p_g	T	$X^g_{CO_2}$	p_g	T	$X^g_{CO_2}$
Two-phase	p_g	T	s_g	p_g	T	p_t	p_g	s_g	$X^l_{CO_2}$

2.1.2. Flash Method. An alternative approach to variable switching is the flash method. Although the variable switching method is often considered stable and efficient [2], it has several shortcomings: (1) it causes perturbations during Newton iterations when phase changes take place; (2) the change in the definition of independent variables affects the structure of the Jacobian matrix; and (3) as a consequence this degrades performance of the preconditioner during the linear solve. Finally, the variable switching approach is not appropriate for use with multilevel solvers because of the possibility for the need to solve for different independent variables on different levels [1].

In the flash approach the primary variables preserved during the solution of the governing equations. The flash method has been implemented in the FLASH2 mode in PFLOTRAN. The primary variables are p, T, and the total mole fraction z_i of the ith component summed over all phases, defined as:

$$z_i = \frac{\sum_\alpha n_i^\alpha}{\sum_\alpha \sum_j n_j^\alpha}, \tag{11a}$$

$$= \frac{\sum_\alpha s_\alpha \rho_\alpha x_i^\alpha}{\sum_\alpha s_\alpha \rho_\alpha}, \tag{11b}$$

where the latter form is derived from the identity

$$\frac{n_i^\alpha}{V} = \frac{n_i^\alpha}{n_\alpha}\frac{n_\alpha}{V_\alpha}\frac{V_\alpha}{V_p}\frac{V_p}{V}, \tag{12a}$$

$$= \varphi s_\alpha \rho_\alpha x_i^\alpha, \tag{12b}$$

where V is the total control volume and V_p denotes the pore volume. Explicitly for a two-phase ($\alpha = w$, SC) system where w designates the phase H_2O and SC designates supercritical CO_2, z_i can be expressed as

$$z_i = \frac{s_w \rho_w x_i^w + s_{\mathrm{SC}} \rho_{\mathrm{SC}} x_i^{\mathrm{SC}}}{s_w \rho_w + s_{\mathrm{SC}} \rho_{\mathrm{SC}}}, \tag{13}$$

for molar fluid densities ρ_w, ρ_{SC}, and saturation s_w, $s_{SC} = 1 - s_w$. The variable z_i is a persistent degree of freedom throughout the simulation.

Let x_i, y_i be the mole fraction of component i in liquid and supercritical phases, respectively, related by the equilibrium constant K_i by the expression

$$y_i = K_i x_i, \tag{14}$$

and let ζ_{SC} represent the supercritical phase mole fraction defined as

$$\zeta_{\mathrm{SC}} = \frac{\sum_i n_i^{\mathrm{SC}}}{\sum_\alpha \sum_i n_i^\alpha}. \tag{15}$$

Under a phase transformation mass conservation implies the relation

$$z_i = \zeta_{\mathrm{SC}} y_i + (1 - \zeta_{\mathrm{SC}}) x_i. \tag{16}$$

Using Eqn. (14) it follows that

$$z_i = \zeta_{\mathrm{SC}} K_i x_i + (1 - \zeta_{\mathrm{SC}}) x_i, \tag{17}$$

from which

$$x_i = \frac{z_i}{1 + (K_i - 1)\zeta_{\mathrm{SC}}}, \tag{18a}$$

and

$$y_i = \frac{K_i z_i}{1 + (K_i - 1)\zeta_{\mathrm{SC}}}. \tag{18b}$$

The value of ζ_v can be found by solving the flash equation (i.e., the Rachford-Rice equation)

$$F(\zeta_{\mathrm{SC}}) = \sum_i (y_i - x_i) = \sum_i \frac{z_i(K_i - 1)}{1 + (K_i - 1)\zeta_{\mathrm{SC}}} = 0. \tag{19}$$

3. Comparison of Variable Switching and Flash Approaches

Variable switching and flash approaches are compared for a 3D injection problem. Physical properties used in the simulations consist of the Span-Wagner [4] EOS for supercritical CO_2 and the density mixture correlation taken from Duan et al. [3]. The variation in mixture density with temperature calculated using this correlation is shown in Figure 1 for different brine concentrations. At higher temperatures the mixture density becomes less than the brine density. However, at lower temperatures the mixture density is greater than the brine density and the mixture sinks [5]. It should be noted that use of ideal mixing results in a mixture density that is always less than the brine density. The solubility of CO_2 in brine was calculated using the correlations presented in Duan et al. [6].

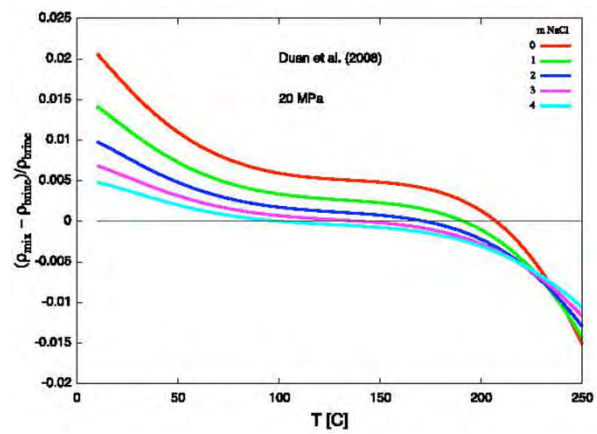


Figure 1. Comparison of mixture density for different brine concentrations for CO_2-brine mixtures of 0, 1, 2, 3, 4 m NaCl with the pure brine density as a function of temperature using the correlation derived in [3].

Performance results for a small 3D problem with domain size 7 km × 7 km × 250 m with grid spacings of $\Delta_x = \Delta_y = 43.75$ m and $\Delta_z = 10$ m are listed in Table 2 with scaling results shown in Figure 2. In these simulations the presence of NaCl was not considered. A formation permeability of

Table 2. Comparison of performance results of variable switching and flash methods for 3D CO_2 injection problem.

Mode	Steps	Newton	Linear	Time-Step Cuts	Time [s]
Var. Switch.	3467	3467	605701	0	4526.2
Flash	2389	2389	408406	0	2973.9

2 Darcy and porosity of 0.38 was used. Injection of supercritical CO_2 at the center of the domain at a rate of 0.2487 kg/s over a period of 25 years. The problem was run on Jaguar XT5 at ORNL using 240 processor cores. For variable switching the first set of variables listed in Table 1 were used. The flash approach shows significant improvement compared to variable switching.

4. Conclusion

The flash and variable switching methods were compared for the two-phase problem of injection of supercritical CO_2 into a brine reservoir fluid. It was found the the flash approach gave better performance compared to variable switching.

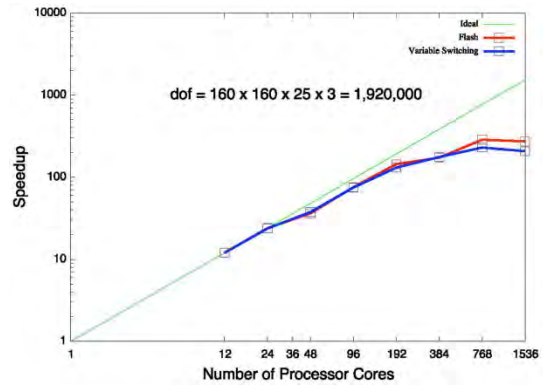


Figure 2. Scaling results for variable switching and flash methods.

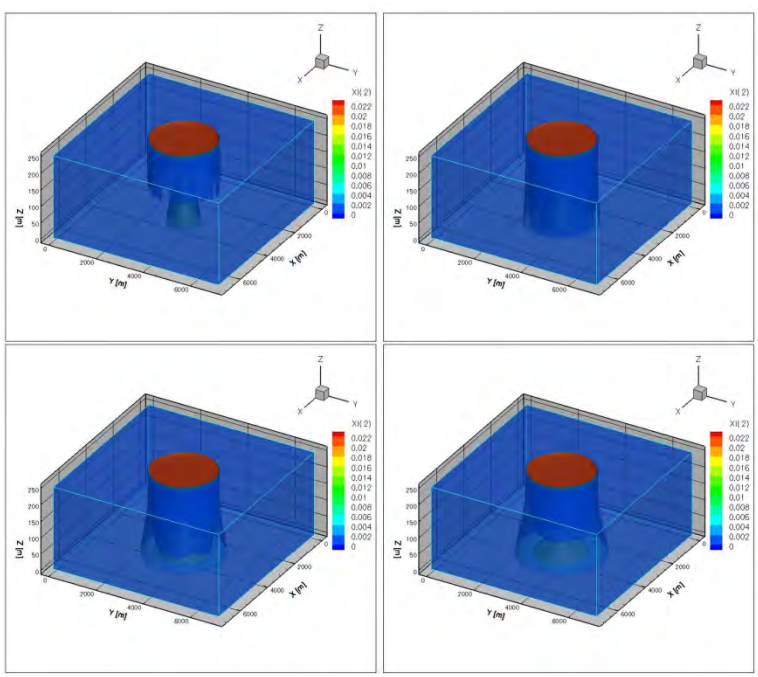


Figure 3. Dissolved CO_2 plotted at times 50, 100, 200 and 300 years.

References

[1] Class H, Helmig R, and Bastian P. 2002. *Advances in Water Resources* **25** 551–564.
[2] Pruess K, Oldenburg C, and Moridis G. 1999. TOUGH2 User's Guide, V2.0 Tech. Rep. Report LBNL-43134, Lawrence Berkeley National Laboratory, Berkeley, CA.
[3] Duan Z, Jia J, Li D, and Mao S. 2008. *Energy & Fuels.*
[4] Span R and Wagner W. 1996. *J. Phys. chem. Ref. Data* **25** 1509–1596.
[5] Lu C, Han W S, Lee S Y, McPherson B J, and Lichtner P C. 2009. *Advances in Water Resources* **32** 1685–1702.
[6] Duan Z and Sun R. 2003. *Chemical Geology* **193** 257–271.

Engineering PFLOTRAN for Scalable Performance on Cray XT and IBM BlueGene Architectures

Richard Tran Mills,[1] Vamsi Sripathi,[2] G. (Kumar) Mahinthakumar,[3] Glenn E. Hammond,[4] Peter C. Lichtner,[5] Barry F. Smith[6]

[1]Computational Earth Sciences Group, Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6015
[2]Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206
[3]Department of Civil, Construction, and Environmental Engineering, North Carolina State University, Raleigh, NC 27695-7908
[4]Hydrology Group, Environmental Technology Division, Pacific Northwest National Laboratory, Richland, WA 99352
[5]Hydrology, Geochemistry, and Geology Group, Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545
[6]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439-4844

E-mail: rmills@ornl.gov, vamsi_s@ncsu.edu, gmkumar@ncsu.edu, glenn.hammond@pnl.gov, lichtner@lanl.gov, bsmith@mcs.anl.gov

Abstract. We describe PFLOTRAN—a code for simulation of coupled hydro-thermal-chemical processes in variably saturated, non-isothermal, porous media—and the approaches we have employed to obtain scalable performance on some of the largest scale supercomputers in the world. We present detailed analyses of I/O and solver performance on Jaguar, the Cray XT5 at Oak Ridge National Laboratory, and Intrepid, the IBM BlueGene/P at Argonne National Laboratory, that have guided our choice of algorithms.

1. Introduction

Subsurface (groundwater) flow and reactive transport models have become important tools for tasks such as understanding contaminant migration and cleanup, assessing geothermal energy technologies, and exploring geologic carbon sequestration as a possible means to reduce greenhouse gas emissions. Although simple groundwater models will suffice in some instances, there there are many in which sophisticated, three-dimensional, multi-scale models employing multiple fluid phases and chemical components coupled through a suite of biological and geochemical reactions are required. The increased complexity of such models demands computing power far beyond that of desktop computers, and in some cases requires the ability to utilize true leadership-class supercomputers.

In this paper, we present analyses of I/O and solver performance in one such complex groundwater simulation code, PFLOTRAN, on two of the most powerful computers in the world—Jaguar, the Cray XT5 at Oak Ridge National Laboratory, and Intrepid, the IBM BlueGene/P at Argonne National Laboratory—and discuss some of the strategies we have employed to achieve parallel scalability on these machines.

2. The subsurface reacting flow simulator, PFLOTRAN

PFLOTRAN [1–6] solves a coupled system of mass and energy conservation equations for a number of phases, including air, water, black oil, and supercritical CO_2, and for multiple chemical components. It utilizes a finite-volume spatial discretization combined with either operator-split or backward-Euler (fully implicit) time stepping. PFLOTRAN is written in Fortran 95 using as modular and object-oriented an approach as possible within the constraints of the language standard, and, being a relatively new code, it is unencumbered by legacy code has been designed from day one with parallel scalability in mind. It is built on top of the PETSc framework [petsc-web-page, petsc-user-ref, petsc-efficient] and makes extensive use of features from PETSc, including iterative nonliner and linear solvers, distributed linear algebra data structures, parallel constructs for representing PDEs on structured grids, performance logging, runtime control of solver and other options, and binary I/O. It employs parallel HDF5 [10] for I/O and SAMRAI [11] for adaptive mesh refinement.

PFLOTRAN employs domain-decomposition parallelism, with each subdomain assigned to an MPI process and a parallel solve implemented over all processes. A number of different solver and preconditioner combinations from PETSc or other packages can be used, but in this paper we employ an outer, inexact Newton method with an inner BiCGStab linear solver or an "improved" variant (described in Section 3.1), preconditioned with a block-Jacobi method employing point-block ILU(0) on each subdomain. Message passing (3D "halo exchange") is required to exchange ghost points across subdomain boundaries, and, within the BiCGStab solver, gather/scatter operations are needed to handle off-processor vector elements in matrix-vector product computations, and global reduction operations are required to compute vector inner products and norms.

3. Scalability on Cray XT and IBM BlueGene Architectures

PFLOTRAN exhibits good parallel scalability on a number of different machine architectures including Jaguar, the Cray XT5 at ORNL, and Intrepid, the IBM BlueGene/P at Argonne, and has been run on up to 131,072 compute cores. Figure 2 illustrates the strong-scaling performance (using the default BiCGStab implementation) observed on the Cray XT5 and the IBM BG/P for the flow and reactive transport solves for a benchmark problem that simulates the release of a hypothetical uranium plume at the Hanford 300 Area in southeastern Washington state. This problem is based on that described in [12]. Our version of the benchmark problem consists of $850 \times 1000 \times 80$ cells and includes 15 primary chemical species, resulting in 68 million degrees of freedom for the flow solve and approximately one billion degrees of freedom for the reactive transport solve (flow and transport are coupled sequentially in this simulation).

Speedup tapers off on the XT5 primarily due to the high cost of allreduce operations on the machine. Figure 1 illustrates the scaling of the time spent in MPI_Allreduce() for the combined benchmark flow and transport problem; the BG/P outperforms the XT5 at 16380 processor cores, which coincides closely with the cross-over point for the flow solve in Figure 2. Unlike the BG/P, the XT5 does not have a separate, tree-structured network for such operations, which partly explains the poorer scalability on the XT5. We believe that operating system noise is also a significant contributor to the poor scalability observed on the XT5, as we have observed significant load imbalance on synthetic MPI_Allreduce() benchmarks that are perfectly load balanced within the application. Figure 3 displays box plots (also known as box-and-whisker plots) summarizing the MPI_Allreduce() timings (including synchronization time) on the Cray XT5 and IBM BG/P. Interpreting these plots is somewhat less than

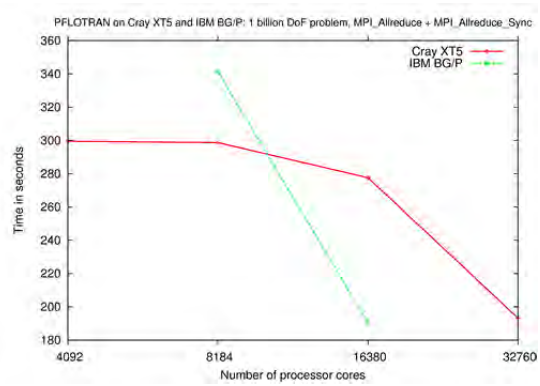


Figure 1. Scaling of time spent in MPI_Allreduce() for the benchmark flow and transport problem.

straightforward because the simulation employs a structured grid containing inactive cells along the river boundary; MPI processes with many inactive cells have very little computation to do, and therefore spend a long time waiting at MPI_Allreduce() calls. These lightly-loaded processes are responsible for the great majority of the outliers (represented by "+" symbols in the plots), although they constitute only about 5% of the processes. Note that these wait times are much shorter on the XT5 because the CPUs are faster, and not because the communication is better.

3.1. Mitigating Cost of Global Reductions in Krylov Solvers

At high core counts (on the Cray XT5, especially), linear solves may be dominated by cost of MPI_Allreduce() calls to calculate vector dot products and norms. We have added a PETSc implementation of the so-called Improved BiCGStab algorithm (IBCGS) [13] that requires only 2 MPI_Allreduce() calls per iteration, instead of the usual 3. If the residual norm calculation is lagged by an iteration, it is possible to wrap everything into a single MPI_Allreduce(), at the cost of doing one additional iteration; we lag this calculation in the flow solve, where a high number of iterations are required, but not in the transport solve, where few linear solver iterations are required at each Newton step. The algorithm is considerably more complicated, requires the transpose matrix vector product (applied only during the first iteration), and performs some extra, local vector operations.

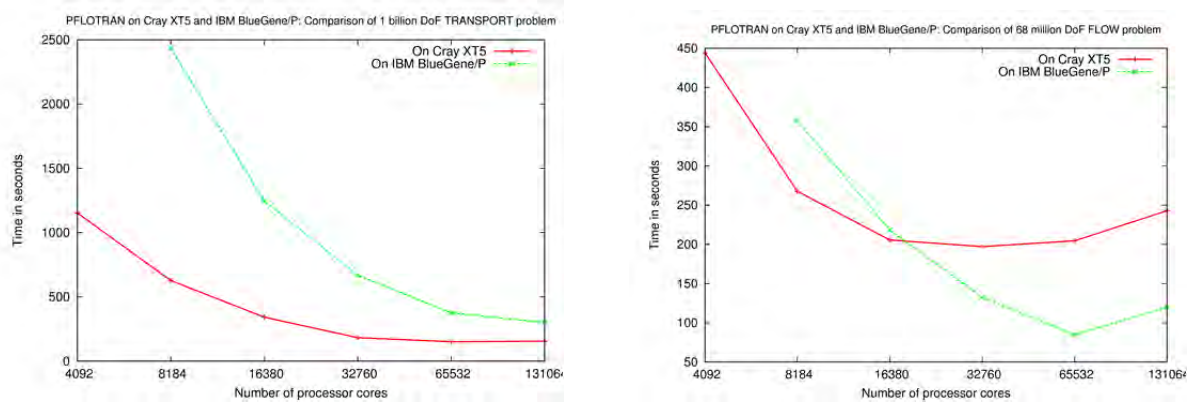


Figure 2. Scaling on the Cray XT5 and IBM BG/P systems for 30 steps of the uranium plume migration problem benchmark. The reactive transport solve (left) has 1 billion degrees of freedom, and the corresponding flow problem (right) has 68 million degrees of freedom. Note that the flow problem is very small for these core counts.

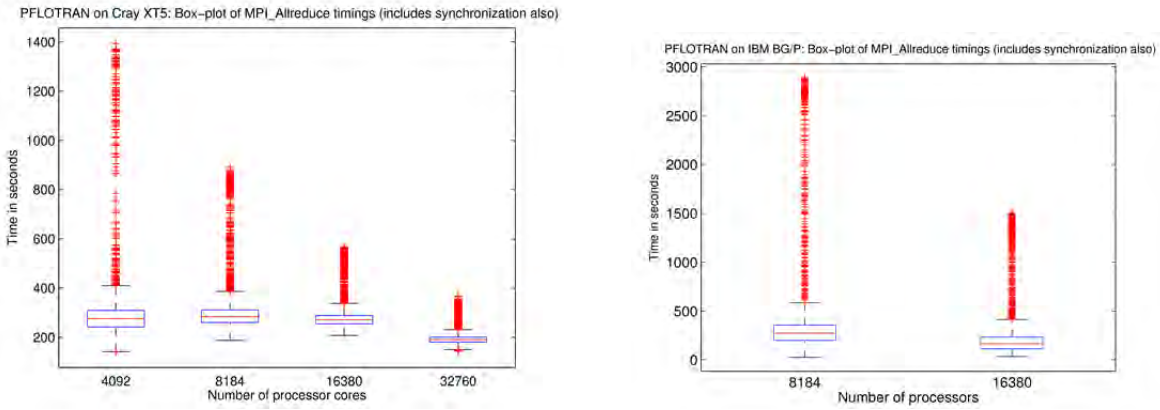


Figure 3. Box-plots of MPI_Allreduce timings (including synchronization time) for the XT5 (left) and BG/P (right) for the flow and reactive transport benchmark problem. Points which cross the whiskers are marked with "+" symbols and represent outliers.

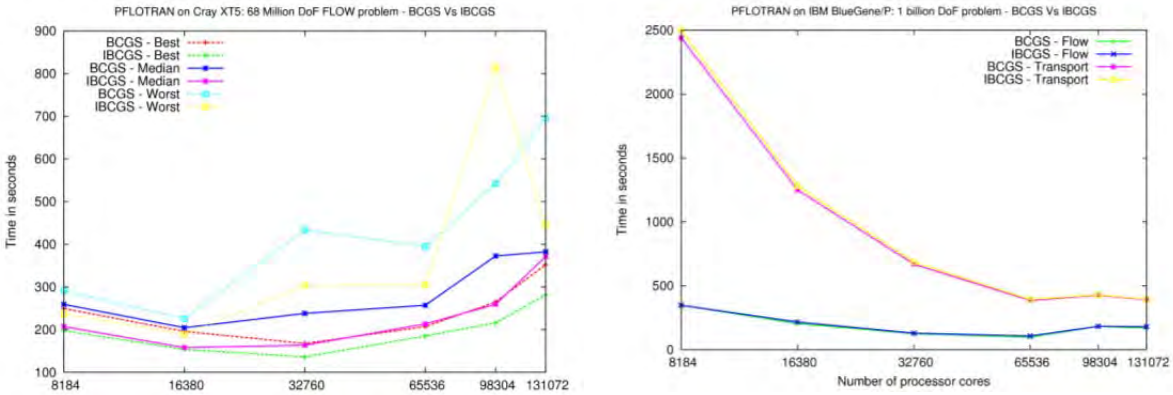


Figure 4. Comparisons of the conventional BiCGStab (BCGS) algorithm and the Improved (IBCGS) algorithm for a 68 million degrees of freedom flow problem on the Cray XT5 (left) and IBM BG/P (right, with performance for accompanying transport problem with 1 billion degrees of freedom also depicted). This is an artificially small problem for these high core counts, so the cost of communication dominates, as there is relatively little computation per core to be done. Due to variability in machine performance as well as IBCGS iteration counts, best, worst, and median times are reported for the XT5 runs.

Despite the somewhat higher operation count, the reduced communication costs can lead to significant performance gains; we have found that the restructured solver is generally faster on the Cray XT4/5 using more than a few hundred processor cores. Figure 4 displays the performance of the conventional and restructured BiCGStab solvers on the Cray XT5 and IBM BG/P. Due to instability within the restructured solver, the number of iterations required to solve the same problem varies between runs. The restructed algorithm offers significant performance improvements on the Cray XT5, but on the IBM BG/P, where global reduction operations exhibit better scalability, we observe little performance improvement.

3.2. Improving Parallel I/O with a Two Phase Approach

PFLOTRAN uses parallel I/O in the form of 1) routines employing parallel HDF5 to read input files and write out simulation output fies, and 2) direct MPI-IO calls in a PETSc Viewer backend to write out checkpoint files. The original HDF5 routines perform well on the IBM BG/P architecture, but on the Cray XT architecture, these routines do not scale to high core counts due to too many small I/O requests, and contention for the small number of Lustre Metadata Servers. The rememdy this problem, members of the SciDAC Performance Engineering Research Institute (G. Mahinthakumar and V. Sripathi) have collaborated with the PFLOTRAN team to implement a two-phase (communication phase and I/O phase) approach [14]. Instead of all processes participating in collective read or write operations, the MPI global communicator is split into sub-communicators, with the root process of each communicator performing I/O for the entire group and appropriate gather/scatters to collect or distribute data to/from group members. We note that MPI-IO implementations generally provide support for two-phase I/O, but we have chosen to implement this at the application level because 1) In the PFLOTRAN initialization phase, not all processes participate in all HDF5 read calls, and 2) Attempting to use the MPI-IO two-phase I/O results in exhaustion of resources in the low-level Portals library used for inter-node communication on the Cray XT architecture. Our improved I/O routines yield **25X improvement** in the initialization phase and **3X improvement** in the write phase at 65,536 cores (quad-core processors) on the Cray XT5.

4. Conclusions and Future Work

The performance of PFLOTRAN is being continuously benchmarked and improved as the code is developed and new machines come online. In this paper we have examined the performance of

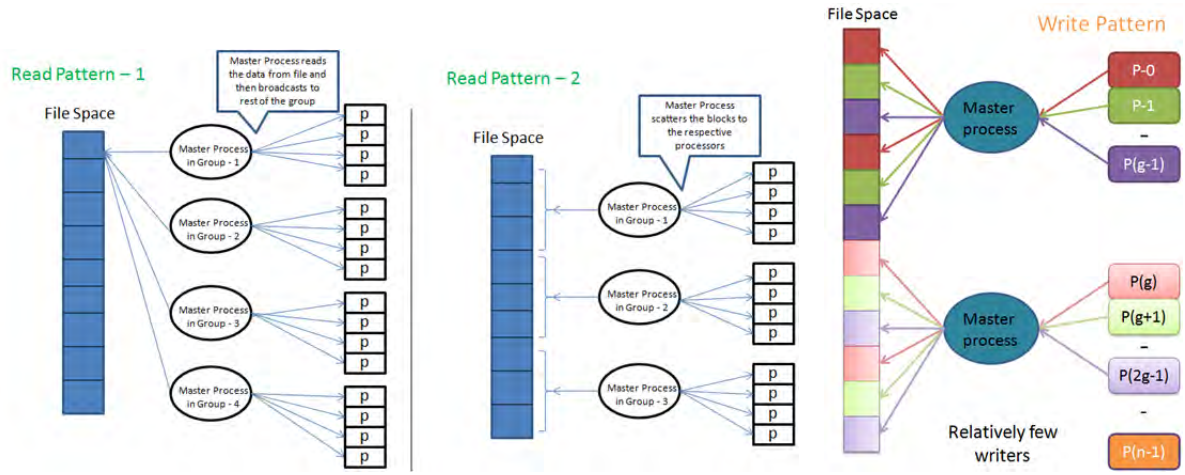


Figure 5. Improved (two phase) read and write patterns. There are two different patterns of reads in the initialization phase.

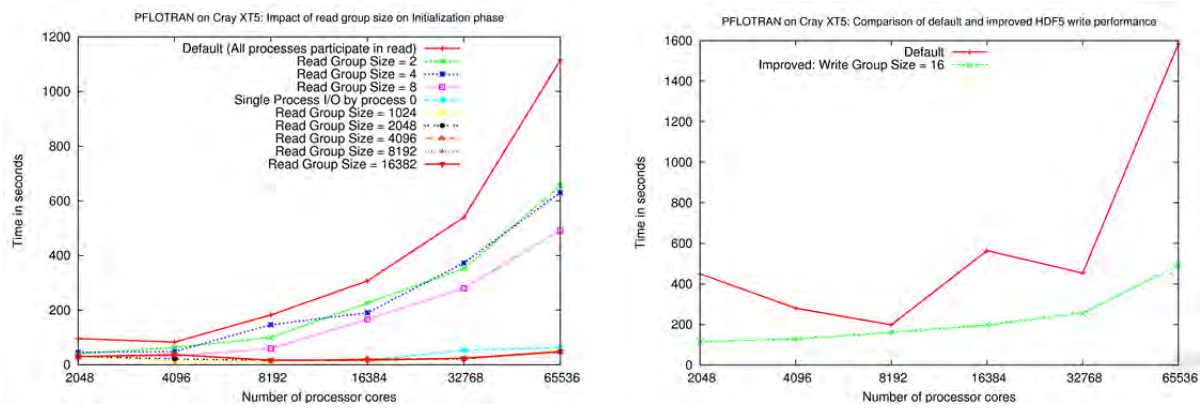


Figure 6. Comparisons of default initialization (read) and write behavior in PFLOTRAN with improved two phase implementations.

PFLOTRAN on two leadership-class machine architectures—the Cray XT5 and IBM BlueGene/P—and discussed algorithmic changes made to solvers (restructured BiCGStab algorithm) and I/O routines (two-phase I/O) to address particular performance bottlenecks we have observed. Such work will continue as as new capabilities (e.g., unstructured grids, structured adaptive mesh refinement, multi-continuum formulations) are developed and as new architectures (e.g., nodes with GPU or other "accelerator" devices) come online. At a more fundamental level than performance tuning, we also will continue to investigate scalable solver and preconditioner algorithms that are well-tailored to our application while also possessing intrinsic scalability.

Acknowledgements
This research was partially sponsored by the Climate and Environmental Sciences Division (CESD) of the Office of Biological and Environmental Research (BER) and the Computational Science Research and Partnerships (SciDAC) Division of the Office of Advanced Scientific Computing Research (ASCR) within the U.S. Department of Energy's Office of Science (SC). This research used resources of the National Center for Computational Sciences (NCCS) at Oak Ridge National Laboratory (ORNL), which is managed by UT-Battelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Pacific Northwest National Laboratory is managed for the U.S. Department of Energy by Battelle Memorial Institute under Contract No. DE-AC06-76RL01830. Argonne National

Laboratory is managed by UChicago Argonne, LLC, for the U.S. Department of Energy under Contract No. DE-AC02-06CH11357. Los Alamos National Laboratory is managed is operated by Los Alamos National Security, LLC, for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396.

References

[1] Mills R T, Lu C, Lichtner P C and Hammond G E 2007 Simulating subsurface flow and transport on ultrascale computers using PFLOTRAN *SciDAC 2007 Scientific Discovery through Advanced Computing (Journal of Physics: Conference Series* vol 78) ed Keyes D (Boston, Massachusetts: IOP Publishing) p 012051

[2] Hammond G E, Lichtner P C and Lu C 2007 Subsurface multiphase flow and multicomponent reactive transport modeling using high-performance computing *SciDAC 2007 Scientific Discovery through Advanced Computing (Journal of Physics: Conference Series* vol 78) ed Keyes D (Boston, Massachusetts: IOP Publishing) p 012025

[3] Lu C and Lichtner P C 2007 High resolution numerical investigation on the effect of convective instability on long term CO_2 storage in saline aquifers *SciDAC 2007 Scientific Discovery through Advanced Computing (Journal of Physics: Conference Series* vol 78) ed Keyes D (Boston, Massachusetts: IOP Publishing) p 012042

[4] Hammond G E, Lichtner P C, Mills R T and Lu C 2008 Towards petascale computing in geosciences: application to the Hanford 300 Area *SciDAC 2008 Scientific Discovery through Advanced Computing (Journal of Physics: Conference Series* vol 125) ed Keyes D (Seattle, Washington: IOP Publishing) p 012051

[5] Mills R T, Hammond G E, Lichtner P C, Sripathi V, Mahinthakumar G K and Smith B F 2009 Modeling subsurface reactive flows using leadership-class computing *SciDAC 2009 Scientific Discovery through Advanced Computing (Journal of Physics: Conference Series* vol 180)

[6] Hammond G E, Lichtner P C, Mills R T and Lu C 2010 *Ground Water Reactive Transport Models* (Bentham Science Publishers) chap PFLOTRAN: Reactive Flow & Transport Code for Use on Laptops to Leadership-Class Supercomputers

[7] Balay S, Buschelman K, Gropp W D, Kaushik D, Knepley M G, McInnes L C, Smith B F and Zhang H 2009 PETSc Web page http://www.mcs.anl.gov/petsc

[8] Balay S, Buschelman K, Eijkhout V, Gropp W D, Kaushik D, Knepley M G, McInnes L C, Smith B F and Zhang H 2009 PETSc users manual Tech. Rep. ANL-95/11 - Revision 3.0.0 Argonne National Laboratory

[9] Balay S, GroppWD, McInnes L C and Smith B F 1997 Efficient management of parallelism in object oriented numerical software libraries *Modern Software Tools in Scientific Computing* ed Arge E, Bruaset A M and Langtangen H P (Birkhäuser Press) pp 163–202

[10] The HDF Group 2009 HDF5 *User's Guide: HDF5 Release 1.8.3* NCSA http://www.hdfgroup.org/HDF5

[11] Wissink A M, Hornung R D, Kohn S R, Smith S S and Elliott N 2001 Large scale parallel structured AMR calculations using the SAMRAI framework *Supercomputing '01: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)* (New York, NY, USA: ACM) pp 6–6 ISBN 1-58113-293-X

[12] Hammond G E and Lichtner P C 2010 Cleaning up the Cold War: Simulating uranium migration at the Hanford 300 Area *Proceedings of SciDAC 2010*

[13] Yand L T and Brent R 2002 The improved BiCGStab method for large and sparse unsymmetric linear systems on parallel distributed memory architectures *Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing* (IEEE)

[14] Sripathi V 2010 *Performance Analysis and Optimization of Parallel I/O in a large scale Groundwater Application on Petascale Architectures* Master's thesis North Carolina State University

Simulations of subsurface flow and transport at the pore and continuum scales

Bruce Palmer,[1] Yilin Fang,[1] Tim Scheibe,[1] and Alexandre Tartakovsky[1]

Pacific Northwest National Laboratory
P.O. Box 999
Richland, WA 99352

E-mail: bruce.palmer@pnl.gov

Abstract. This paper presents recent results of simulations of subsurface and flow using both pore scale and grid-based continuum methods. Large scale simulations of pore scale flow and transport using the Smoothed Particle Hydrodynamic method (SPH) using more than 14 million particles have been performed. These simulations are focused on tracking the behavior of contaminants and multiphase fluids in the subsurface. Simulations are also being performed at the field scale using the Subsurface Transport Over Multiple Phases code (STOMP) using a novel, highly-resolved synthetic model of a braided channel belt deposit to analyze the large scale transport of material in a deposit with multiscale heterogeneity. Both sets of simulations are being performed with codes that have been developed using component-based software engineering principles that result in highly modular codes with few dependencies between components.

1. Introduction

Groundwater simulations encompass a range of scales, ranging from atomic and molecular scale processes occurring at mineral interfaces to processes at the field scale that can cover tens to hundreds of kilometres. This project has focused on developing methodologies for applying large scale simulations at the pore and continuum scales, with the goal of eventually coupling across these scales using a hybrid simulation approach. The pore scale simulations are based on the Smoothed Particle Hydrodynamics method (SPH), which is a particle based approach to solving the hydrodynamic equations [1]. SPH has the advantage that it can easily be adapted to solving for flow in the topologically complex geometries found in porous media. It can also be modified to handle processes like multi-phase flow. The continuum simulations are based on the Subsurface Transport Over Multiple Phases code (STOMP) developed at Pacific Northwest National Laboratory for performing lab and field scale simulations of subsurface reactive flow and transport [2].

Both simulation codes are being developed using a component-based software engineering approach that is designed to keep the code modular and minimize dependencies between different components. This reduces the overall complexity of the code and simplifies incorporation of new libraries and algorithms. The component-based versions of these codes were originally developed using the Common Component Architecture toolkit (CCA) [3]. The CCA toolkit provides language interoperability [4] and runtime configuration capabilities that are useful in managing multiple components that may have originally been developed using different languages. It also enforces separate namespaces in a way that encourages developers to keep components highly encapsulated.

This is critical for keeping components from developing significant dependencies via shared header files and modules. The communication in both codes is implemented using the Global Arrays toolkit (GA) [6]. The one-sided communication in GA supports a shared memory style programming model that dramatically simplifies much of the parallel portions of the two codes.

2. Pore Scale Simulations using the Smoothed Particle Hydrodynamics Method

Smoothed Particle Hydrodynamics is a Lagrangian method for solving the equations describing fluid flow. It is based on the idea that a smooth approximation A_S to a continuum field A can be generated from a discrete sampling of the field in space A_i using the formula

$$A_S(\mathbf{r}) = \sum_i W(\mathbf{r} - \mathbf{r}_i) A_i$$

where $W(\mathbf{r})$ is a normalized weighting function that is at least localized around zero and may be exactly zero outside some finite value of $|\mathbf{r}|$. This equation can be used as the basis for developing approximations to the gradients and higher order derivatives of the continuum field and these, in turn, can be used to write approximate equations of motion for the field. Because the set of points $\mathbf{r}_i$ at which the field is evaluated are carried through the system by local values of the velocity field, they behave like particles and many of the numerical techniques developed for particle simulations can be applied to fluid simulations using the SPH method.

A parallel code based on a spatial decomposition algorithm has been developed as part of this project to simulate the behaviour of millions of particles in 2 and 3 dimensions. The code framework is built around a data manager that provides a mechanism for annotating large data objects in the code so that any component can determine whether it can use or modify a block of data. If a component recognizes a piece of data, it caches a pointer to the data during an initialization step. The data can then be accessed in the remainder of the code using conventional array dereferencing methods, so data access is fast. The advantage of annotating the data is that there is no need to explicitly pass data through argument lists or to share global variables. This simplifies the interfaces of components enormously and eliminates most of the dependencies that mean that a change to one component implies numerous changes to other components. A complete description of the framework is available in [7].

The current SPH framework contains several force components and a simple chemistry component that can be used to simulate transport of a non-reactive contaminant. A wiring diagram for a configuration of the SPH framework using the CCA toolkit is shown in Figure 1.

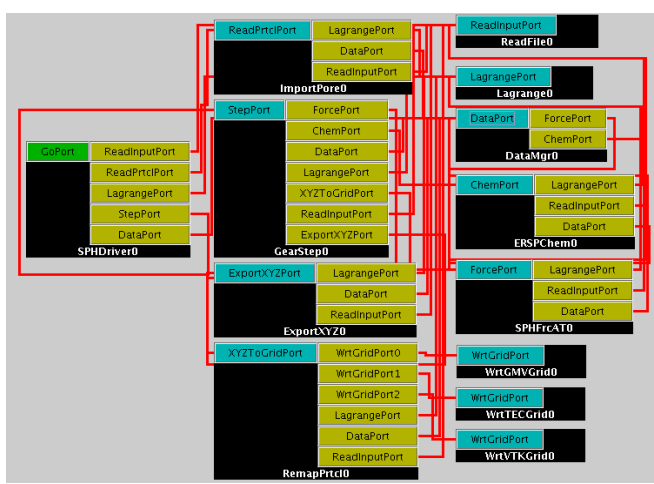


Figure 1. Schematic diagram of SPH application built from components.

The existing CCA-based framework is being used for a number of large scale calculations. Some preliminary calculations have been done in two dimensions on an experimental configuration being used to investigate multiphase flow in porous media. The experiment consists of a large array of cylinders between two flat plates. Preliminary numerical tests are in progress to determine if detailed simulations of this system using SPH are feasible. These tests are currently using about 14 million particles on 2048 processes. Plots of the cylinder array test

configuration and the velocity field from an SPH simulation are shown in Figure 2. Additional simulations are planned using terms that account for the thin film nature of the flow.

The SPH code has also been used to investigate multiphase flow in model porous media..A plot of a simulation of two-phase flow in a synthetic porous medium generated through a Monte Carlo algorithm is shown in Figure 3. A non-wetting fluid is entering the bottom of the figure and displacing a wetting fluid. The figure shows fingering of the invading fluid as it tries to displace the wetting fluid coating the solid surfaces.

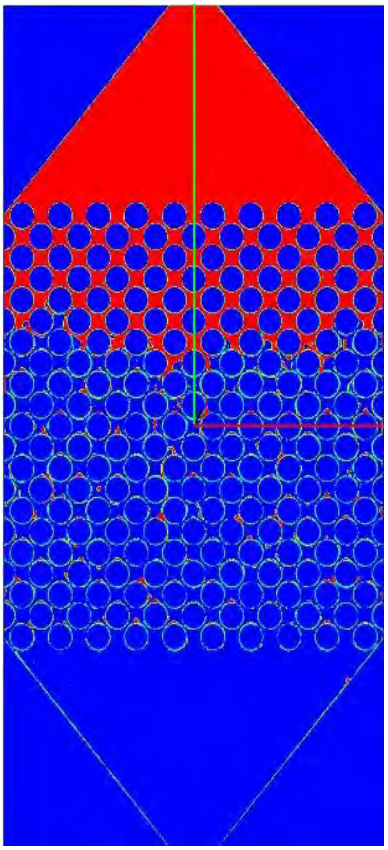


Figure 2. Graphic of simulation of a ½ scale experimental configuration of cylinders used to investigate two-phase flow in porous media. The solid material is shown in blue, the displacing fluid is in red.

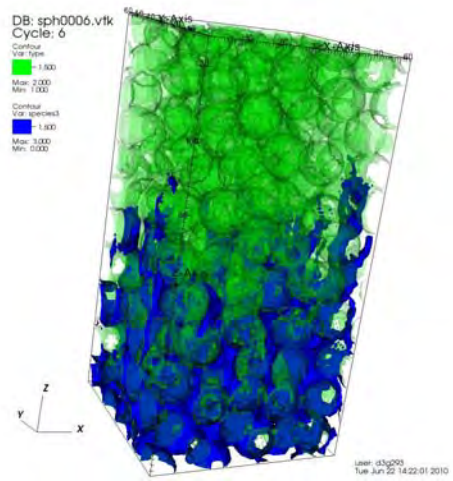


Figure 3. Simulation of invasion of a non-wetting fluid (blue) that is displacing a wetting fluid (not shown). The solid matrix representing the porous media is shown in transparent green.

3. Darcy Scale Simulations using STOMP

Continuum simulations based on Darcy scale models of flow and transport are used in most subsurface modelling applications. Because of their non-linearity, overall complexity, and tight coupling, these models still present great numerical challenges. Work has been underway to decompose the STOMP subsurface simulator into components using the CCA framework. Many of the strategies used in developing the SPH framework described above are also being employed for STOMP. The decomposition has been based on a data model that treats the numerical grid in STOMP as an unstructured grid. This approach was chosen to enable the future use of unstructured grids. The current

STOMP code has been split into several components. The first is a grid component that initializes and maintains the grid geometry. The grid component can also add and delete fields to the grid and is responsible for updating the ghost cells of fields on the grid when requested to do so by other components. Most of the remaining physics and numerical routines are contained in a second large component, but this has been targeted for further decomposition. Additional components include the initial setup (import), mesh IO (output), and a data manager. The solver routines, chemistry, and time integration routines have also been identified as future targets for componentization. A schematic of the existing STOMP framework is shown in Figure 4.

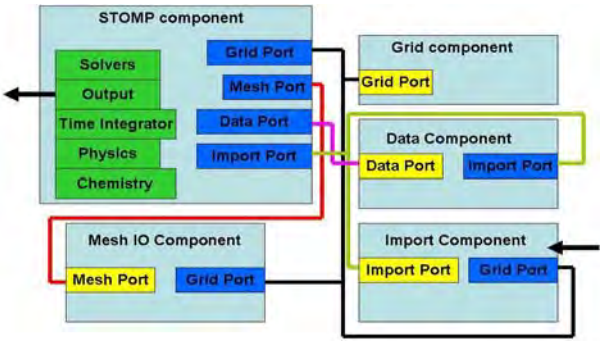


Figure 4. Schematic diagram of STOMP framework showing the main physics driver, grid component, mesh IO component, import component, and data manager. Additional potential components are denoted by green boxes. The interfaces defining components are shown by the blue (uses) and yellow (implements) boxes.

The component-based version of STOMP has been used for a number of large simulations. An example is a calculation of flow and transport in a synthetic braided channel belt deposit based on a model by Ritzi [8] that involved 39 million grid cells and was run using up to 4096 processors. A representation of the braided deposit is shown in Figure 5. The flow simulations derived from this geologic structure were subsequently analyzed using particle tracing methods to generate the visualization of the flow field seen in Figure 6. These analyses are being used to determine what procedures can be used to integrate out the fine scale structure from these systems so that larger regions can be simulated using coarser resolution meshes. These upscaling methods are crucial to developing simulation methodologies that can model the behaviour of large regions over hydrologically significant timescales.

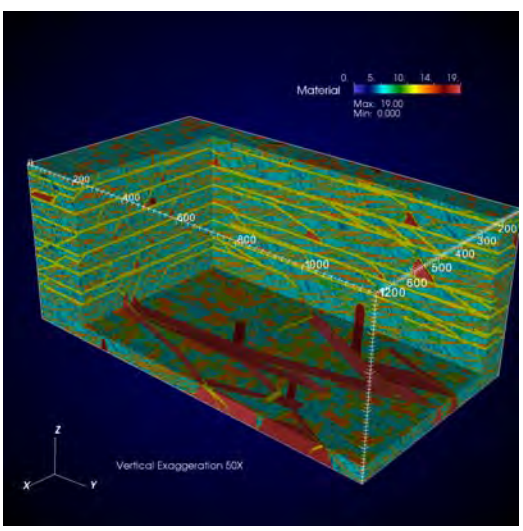


Figure 5. Synthetic model of a braided channel belt deposit following Ritzi [8].

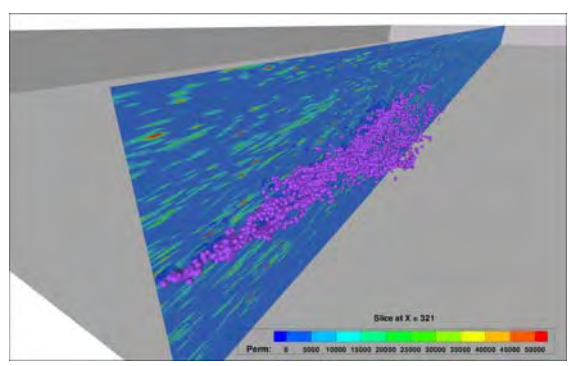


Figure 6. Tracers following flow field using the Ritzi configuration shown in Figure 6.

4. Summary

The present work has demonstrated that a component-based approach can be used to develop groundwater simulation codes that maintain high performance and can be used to solve a variety of challenging subsurface simulation problems. Further work will focus on systematic studies of flow at the pore scale to develop a detailed picture of the behaviour of two-phase flows in porous media. Simulations of reactive transport in porous media, focusing particularly on uranium transport, are also planned. Continued work on the STOMP code will focus on further decomposition of this code into additional components as well as including additional functionality into the STOMP framework. A primary focus will be to include a CO_2 modelling capability that can be used for simulations of carbon sequestration in various geologic formations.

Acknowledgments

This research is supported by the U. S. Department of Energy's Office of Science under the Scientific Discovery through Advanced Computing (SciDAC) program. A portion of this research was performed using the Molecular Science Computing Facility at EMSL, a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at Pacific Northwest National Laboratory.

References

[1] Monaghan, JJ (2005). *Smoothed particle hydrodynamics Rep. Prog. Phys.*, **68** 1703–1759.

[2] White, MD, and M Oostrom (2000). STOMP Subsurface Transport Over Multiple Phases Version 2.0 Theory Guide, PNNL-12030, 235 pp., Pacific Northwest Natl. Lab., Richland, Wash.

[3] White, MD, and M Oostrom (2006). STOMP Subsurface Transport Over Multiple Phases Version 4.0 User's Guide, PNNL-15782, 120 pp., Pacific Northwest Natl. Lab., Richland, Wash.

[4] Allan BA, R Armstrong, DE Bernholdt, F Bertrand, K Chiu, TL Dahlgren, K Damevski, WR Elwasif, TGW Epperly, M Govindaraju, DS Katz, JA Kohl, M Krishnan, JW Larson, S Lefantzi, MJ Lewis, AD Malony, LC McInnes, J Nieplocha, B Norris, SG Parker, J Ray, S Shende, TL Windus, and SJ Zhou (2006). A Component Architecture for High-Performance Scientific Computing, *Int. J. High Perf. Comput. Applications*, **20** 163–202.

[5] Kohn, S, G Kumfert, J Painter, and C Ribbens, Divorcing Language Dependencies from a Scientific Software Library, 10th SIAM Conference on Parallel Processing, Portsmouth, VA, March 12–14, 2001.

[6] Nieplocha, J, B Palmer, V Tipparaju, M Krishnan, H Trease, and E Apra (2006). Advances, Applications and Performance of the Global Arrays Shared Memory Programming Toolkit, *Int. J. High Perf. Comput. Applications*, **20** 203–231.

[7] Palmer, B, V Gurumoorthi, T Schiebe, and A Tartakovsky (2010). A Component-Based Framework for Smoothed Particle Hydrodynamics Simulations of Reactive Fluid Flow in Porous Media, *Int. J. High Perf. Comput. Applications*, **24** 228–239.

[8] Ramanathan R, A Guin, RW Ritzi, DF Dominic, VL Freedman, TD Scheibe, and IA Lunt (2010). Simulating the Heterogeneity in Braided Channel Belt Deposits: Part 1. A Geometric-Based Methodology and Code, *Water Resources Research*, **46** W04515.

Task parallel sensitivity analysis and parameter estimation of groundwater simulations through the SALSSA framework

Karen Schuchardt[1], Khushbu Agarwal[1], Jared Chase[1], Mark Rockhold[1], Vicky Freedman[1], Todd Elsethagen[1], Tim Scheibe[1], George Chin[1], Chandrika Sivaramakrishnan[1]

[1]Pacific Northwest National Laboratory, PO Box 999 Richland WA 99352

E-mail: karen.schuchardt@pnl.gov

Abstract. The Support Architecture for Large-Scale Subsurface Analysis (SALSSA) provides an extensible framework, sophisticated graphical user interface, and underlying data management system that simplifies the process of running subsurface models, tracking provenance information, and analyzing the model results. Initially, SALSSA supported two styles of job control: user directed execution and monitoring of individual jobs, and load balancing of jobs across multiple machines taking advantage of many available workstations. Recent efforts in subsurface modelling have been directed at scaling simulators to take advantage of leadership class supercomputers. We describe two approaches, current progress, and plans toward enabling efficient application of the subsurface simulator codes via the SALSSA framework: automating sensitivity analysis problems through task parallelism, and task parallel parameter estimation using the PEST framework.

1. Introduction

Groundwater modeling is inherently uncertain due to our limited knowledge of the subsurface, being hidden from direct observation and for which only very small samples can be taken from a small number of locations, usually at significant cost. In addition, the subsurface contains a high degree of complexity formed over a very broad range of time scales with diverse geophysical and geochemical processes acting on a wide range of length scales [1]. Because of the uncertainty associated with developing a conceptual model of the subsurface, large numbers of simulations are often used to ascertain the sensitivity of the model to certain parameters, gain insights into how a model is behaving or to validate the model against a benchmark problem. These simulations may be conducted as a sensitivity analysis where a parameter space that is known *a priori* is evaluated or through parameter estimation techniques that algorithmically optimize a parameter space under user specified conditions or constraints.

Sensitivity analysis and parameter estimation are popular techniques in groundwater modeling. However these can be time consuming to conduct and manage when applying sophisticated three-dimensional models that require large numbers of processors per simulation. We are developing high-level tools to enable users to fully leverage task parallelism on supercomputers with thousands to tens of thousands of processors. In this paper, we describe ongoing extensions to our groundwater simulation software environment that enable users to easily generate and run sensitivity analyses and our plans for supporting parameter estimation while maintaining full simulation provenance records.

The remainder of this paper is organized as follows: section 2 provides brief background information on our software environment and recent extensions; section 3 describes current developments to support sensitivity analysis applied to the Smoothed Particle Hydrodynamics (SPH) code [2], and section 4 describes our approach to supporting parameter estimation using Parameter Estimation Toolkit (PEST) [3] and the parallel flow and transport code Subsurface Transport Over Multiple Phases (STOMP) [4].

2. Background

The *Support Architecture for Large-scale Subsurface Simulation and Analysis* (SALSSA) software is a user environment for running simulations, tracking inputs and outputs of each simulation, recording provenance information, and accessing analysis and visualization tools. It can be thought of as both an activity tracking system and a dashboard to provide summary and detailed information about individual activities. While the framework was created for groundwater modeling, its design is generic and can be applied to simulators from any number of scientific domains. The key components are a content store for tracking artifacts associated with running and analyzing simulations (input, outputs, parameters), a provenance recording and display capability, a job launching component for launching simulations from the desktop to virtually any computer, a tool registry for adding new tools to the framework, and a baseline set of tools constructed specifically for the SPH and STOMP codes [5-6]. A number of capabilities have recently been added to the SALSSA framework and are summarized below.

Archive: Because simulations can generate data set sizes in the hundreds of gigabytes and beyond, we have added the capability for the user to selectively archive simulations. The content store maintains metadata about the outputs including references to their physical location. An "archive" is any machine that accepts files over one of the standard file transfer protocols. Users can add new archives to the system through a simple registration mechanism.

Reconnect: A "reconnect" function within the workflow enables users to reconnect to a running (or completed) simulation that was initiated and then disconnected from SALSSA due to network errors, computer shutdown (e.g., while on travel), etc.

Generic workflow Components: Generic workflow components have been developed to simplify building scientific workflows in Kepler. Two such components are the *Data Transfer Actor* which can be used to move files between systems over a variety of protocols and the *Job Launching Actor* which integrates a series of steps in the job execution process [7]. These actors have been incorporated into our workflows resulting in less complex workflows that are adaptive to more operating environments.

Command line tools: Command line tools have been developed for selected steps, such as job submission, that were previously accessed only through the user interface.

3. Sensitivity Analysis

Sensitivity analysis is typically conducted by running a suite of similar problems in which there are no dependencies between the simulations and the full set of desired runs is known *a priori*. These embarrassingly parallel simulations are frequently run on workstations or clusters but there is an increasing recognition that supercomputer task parallel techniques can be applied to run multiple simulations in parallel. In SALSSA, our objective is to make this operation transparent across a variety of systems.

For our representative use case, we selected a series of SPH simulations where the parameters of viscosity, speed of sound, and gravity are varied. Similar runs had previously been completed individually as part of a model validation activity [8]. To add this capability to SALSSA, a number of extensions were developed and are briefly described below.

3.1. Generating Many Simulations

A new capability, designed to support the current practice of planning and documenting sensitivity analyses using spreadsheets, reads a standard Excel spreadsheet and a template input file and generates a series of simulations (Figure 1) which can then be run using the new task parallel job launcher. This provides the user with full control over which simulations should be generated. The column labels in the spreadsheet control value substitution. For example, the viscosity column values will be substituted into the viscosity field in the input template. This capability is available as both a command line and interactive tool.

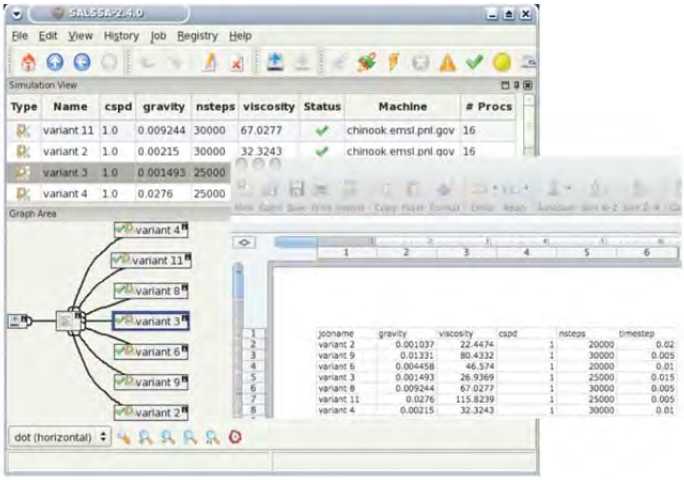


Figure 1. Excel spreadsheet used to define a set of task parallel SPH simulations and the resulting process objects within SALSSA. This spreadsheet contains a combination of settings where gravity and viscosity are varied. Units are dimensionless and specified in terms of the cutoff radius, particle mass, and speed of sound as defined by the SPH code.

3.2. Task Parallel Job Submission and Monitoring

Job submission, monitoring and all data transfer is implemented by a workflow developed within the Kepler Workflow system. The existing job submission components within Kepler have been modified to support multi-task job submissions and to record provenance information associated with the each task within the job. The workflow (Figure 2) is responsible for generating temporary directories for each task and staging files to each tasks' directory (2a), submitting the job (2b), monitoring the job and individual tasks (where possible) (2c), and recording metadata about the job/tasks (2d) and associated output files (2e).

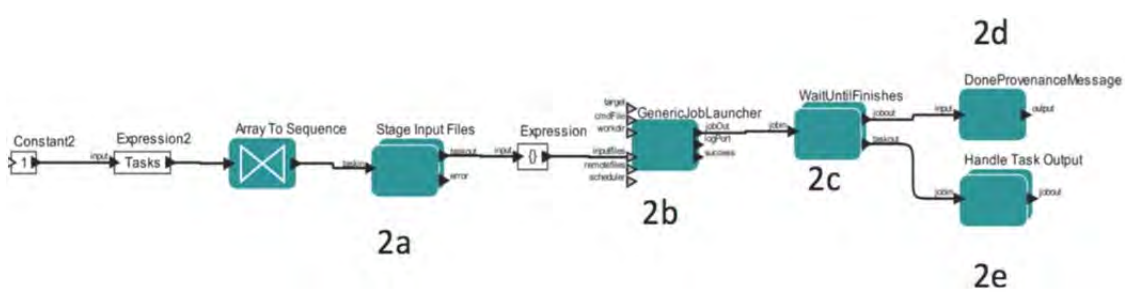


Figure 2. Kepler workflow for executing multiple parallel tasks within a single job submission. Each of the components shown is itself a composite component containing its own sub-workflow. This workflow relies on the Kepler Data Transfer Actor, which is capable of staging files from multiple machines (e.g., desktop and archive).

3.3. Job Scripts

Scheduling on modern supercomputers is generally managed by a combination of workload manager (e.g., MOAB) and resource manager (SLURM, PBS) software. Most of these support the capability of scheduling multiple tasks in a single allocation. The job script for running a task parallel job will

depend on the scheduler and resource manager being used by a particular system. We present here the job scripts corresponding to a MOAB-SLURM system running at the Environmental and Molecular Sciences Laboratory (EMSL) (Figure 3b), and for PBS jobs running at the National Energy Research Scientific Computing Center (NERSC) (Figure 3a). MOAB-SLURM supports task parallel job execution, with built-in commands to monitor and control execution of each task independently. PBS does not have inherent support for task control and monitoring.

```
#PBS -q debug
#PBS -l mppwidth=32
#PBS -l walltime=0:30:00
#PBS -j oe

cd $PBS_O_WORKDIR

for dir in dir0 dir1 dir2 dir3
do
        cd $dir
        aprun –n 8 $cca/local/bin/ccafe-batch --ccafe-rc  sph & >sph.out &
        cd ..
done
wait
```

Figure 3a

```
## Nodes to use for all tasks
#MSUB -l "nodes=4:ppn=8"
....
nodesTask=1
nodeId=0
for dir in dir0 dir1 dir2 dir3
do
      cd $dir
      mpirun -srun -n 8  -N 1 -r $nodeId -o job.out -e job.err $cca/local/bin/
        ccafe-batch --ccafe-rc  sph & > sph.out&
      let "nodeId += $nodesTask"
      cd ..
done
wait
```

Figure 3b

Figure 3. Job script for Cray system running PBS (Franklin) (3a) and HP cluster running MOAB/SLURM(3b).

The general structure of task-parallel jobs involves allocating enough nodes to ensure tasks do not interfere with each other. This also ensures jobs are scheduled on independent sets of nodes. Since job wall-clock time is governed by the longest running task, the user should make certain that each task in the job requires a similar amount of time to run, so as not to waste resources. For simplicity reasons we can assume tasks in the job are square, executing the same code and if run on same number of nodes, will finish in nearly equal time. The job scripts are generated dynamically based on a user-defined set of simulations such as those specified in the spreadsheet of Figure 1 and a target machine and job options are given at the command line or through a user interface.

4. Parameter Estimation

For parameter estimation, our target use-case was a Hanford 300 Area modeling problem which, when run using a serial version of STOMP and the PEST framework, required a total of 364 forward runs to optimize 7 parameters using 7 PEST slaves. This problem took six weeks of continuous computation to complete. Our objective is to apply PEST to a parallel version of STOMP while also modifying PEST so that the slaves are subtasks within a job on the supercomputer. Based on some initial benchmarking of parallel STOMP on 32 processors, we estimate that by using on the order of 200 processors (32 per slave), the same problem could be completed in approximately 30 hours.

PEST may dynamically adapt its approach to computing derivatives. For example, depending on the configuration, it may use a forward difference method requiring one run per parameter and then switch to a central difference method, requiring two forward runs per parameter. PEST may also interrupt its normal processing to focus on a single parameter. Because of these types of adaptive behavior, there may be idle processors and time estimates are approximate. We are currently investigating environments.

5. Summary

We have extended the capabilities of both the SALSSA environment and the Kepler workflow system to enable the parallel execution of multiple, simultaneous parallel simulations on two different supercomputer platforms. This capability enables users to run sensitivity analyses more quickly than can be done in cluster environments. The components developed are general in nature ranging from a spreadsheet based user interface to a generic job launching and job script generation capability along

with workflow components for multi-task job submissions. In addition, we have begun to investigate the adaptation of the PEST framework to support efficient parameter estimation within a single job on a supercomputer. Efficient use of cycles is a key challenge requiring further investigation, although we believe this approach should be effective for a broad class of problems and is being driven by the need to reduce simulation time from months to days for complex 3-dimensional models.

Acknowledgments
This research is supported by the U. S. Department of Energy's Office of Science under the Scientific Discovery through Advanced Computing (SciDAC) program. A portion of this research was performed using the Molecular Science Computing Facility at EMSL, a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at Pacific Northwest National Laboratory. The work was conducted at the Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy under contract DE-AC05-76RLO1830.

References
[1] Scheibe, T., P. Meakin, P. Lichtner, and D. Zachmann, “Science for Problems under the Surface,” SciDAC Review, Issue 7, Spring 2008.
[2] Palmer, B. J., V. Gurumoorthi, A. M. Tartakovsky, and T. D. Scheibe, “A component based framework for smoothed particle hydrodynamics simulations of reactive fluid flow in porous media,” *International Journal of High Performance Computing Applications*, 24(2):228-239, 2010.
[3] PEST Model Independent Parameter Estimation, 5th Addition with updates for 2010. http://www.pesthomepage.org/getfiles.php?file=pestman.pdf
[4] STOMP, http://stomp.pnl.gov , June 2010.
[5] Chase, J. M., K. L. Schuchardt, G. Chin, Jr., J. A. Daily, and T. D. Scheibe. “Iterative workflows in numerical simulations,” In proceedings of IEEE - *The Second International Workshop on Scientific Workflows (SWF ’08)*, Hawaii, July 8, 2008.
[6] Schuchardt, K.L., G. D. Black, J. M. Chase, T. O. Elsethagen and L. Sun, 2007. Process integration framework for subsurface simulations, *Journal of Physics: Conference Series* 78:012047
[7] Ngu, A.H.H., Jamnagarwala, A., Chin, G. Jr., Sivaramakrishnan, C. Critchlow, T. 2010. Context-aware scientific workflow systems using Kepler. Int. J. Business Process Integration and Management, Vol. 5, No. 1, 2010.
[8] Schuchardt KL, JM Chase, JA Daily, TO Elsethagen, BJ Palmer, and TD Scheibe. 2009. “Application of SALSSA Framework to the Validation of Smoothed Particle Hydrodynamics Simulations of Low Reynolds Number Flows.” Journal of Physics: Conference Series 180 (2009), 012065.

Advances in large-scale computations of multi-particle beam dynamics

J. Amundson,[1] J. Cary,[2] P. Lebrun,[1] A. Macridin,[1] L. Michelotti,[1] C. S. Park,[1] P. Spentzouris,[1] P. Stoltz,[2] J. Qiang,[3] J.-L. Vay,[3] and S. Veitzer[2]

[1]Fermi National Accelerator Laboratory, Batavia, IL, USA
[2]Tech-X, Boulder, CO, USA
[3]Lawrence Berkeley National Laboratory, Berkeley, CA, USA

E-mail: amundson@fnal.gov

Abstract. The design and operation of modern particle accelerators is making ever-increasing demands on computer simulations of beam dynamics. We describe recent large-scale computations of collective effects utilizing high-performance computing to address the problems posed by the leading accelerators in operation today and upgrades to these accelerators and facilities planned for tomorrow. We discuss electron cloud, beam-beam and space charge simulations of the Fermilab Tevatron, Main Injector and Debuncher, as well as CERN's LHC, SPS and proposed PS2. In addition, because accelerator science serves the larger purpose of basic physics research, we also describe some of the physics topics that will become accessible due to the upgrades to accelerators we simulate.

1. Introduction

Under the SciDAC program, the Community Petascale Project for Accelerator Science and Simulation (ComPASS) [1] has developed a suite of beam dynamics simulation tools to address the modeling needs of accelerator physicists working to design tomorrow's machines as well as optimizing the machines in operation today. The beam dynamics topics that require high-performance computing are generally collective effects; single-particle effects are important to basic accelerator modeling, but do not on their own require high-performance computing to address. Collective effects, however, require the application of modern, massively parallel computing platforms. We will focus here on three of the most important collective effects in accelerator physics: electron cloud, beam-beam interactions and space charge.

In these proceedings we will describe several ComPASS-based modeling activities, focusing on current and future machines at Fermilab and CERN. We emphasize that these simulations are not done in isolation. Instead, they are each portions of larger design and optimization efforts. Information flows from the simulations to the accelerator physicists in the form of results and, in turn, flows back again in the form of experimental information and new designs. We will highlight four ComPASS codes: VORPAL, BeamBeam3d, IMPACT and Synergia. In addition, we will describe some of the particle physics topics supported by the accelerators targeted by these simulation efforts in order to provide context for the work.

2. Particle Physics at the Frontier(s)

The Particle Physics Project Prioritization Panel (P5) [2] has identified three frontiers in particle physics: the Energy Frontier, the Intensity Frontier and the Cosmic Frontier. Of these, only the Cosmic Frontier does not depend on advances in accelerator technology. The Energy Frontier is the area most commonly discussed in conjunction with particle physics. It is where the largest, most expensive machines are built and operated, and holds out the hope of the most dramatic new discoveries. The Intensity Frontier has historically received less coverage in, e.g., the popular press, but it is an equally important cornerstone of the particle physics program. The highest intensities enable, among other thing, the highest precision measurements and the possibility of discovering new physics at energy scales that will not be directly accessible to the Energy Frontier for at least another generation. In an effort to bring some of the excitement of the Intensity Frontier to a wider audience, we will emphasize it here, leaving a discussion of the Energy Frontier to the already extensive public literature.

Fermilab is proposing to dramatically extend the Intensity Frontier with Project-X [3], which includes a new superconducting 8 GeV linac as well as upgrades to existing Fermilab accelerators to handle the higher intensities demanded by the program. These upgrades require, of course, a better understanding of intensity-dependent effects, and as such are a bread-and-butter aspect of the ComPASS project's beam dynamics efforts. The physics opportunities afforded by Project-X include support for the long-baseline neutrino program, rare-decay searches and high-precision experiments. Ultimately, these experiments will help us address questions about the origins of mass, the observed family structure of particles and the matter-antimatter asymmetry of the universe.

We will highlight one proposed Intensity Frontier experiment at Fermilab, Mu2e. The Mu2e experiment will search for decays of muons into electrons. Such decays are forbidden at tree-level in the Standard Model. In fact, the only way such decays can proceed in the Standard Model is via the so-called "penguin" diagram of Figure 1. The rate for such decays is extraordinarily small. Although the figure contains a virtual photon, the photon can also be real, resulting in $\mu \to e\gamma$ decays. In this latter case, the branching ratio is given by

$$Br(\mu \to e\gamma) = \frac{3\alpha}{32\pi}\left|\sum_{i=2,3} U_{\mu i}^{*} U_{ei} \frac{\Delta m_{1i}^{2}}{m_{W}^{2}}\right|^{2} < 10^{-54}, \tag{1}$$

a rate so small that it is completely beyond the reach of any conceivable experiment. Any observation of $\mu \to e$ transitions, therefore, would be a gold-plated signal of physics beyond the Standard Model. The Mu2e experiment intends to improve the existing limit of $\sim 10^{-11}$ by more than four orders of magnitude before the Project-X era, and another two orders of magnitude during Project-X. In the end, Mu2e will require on the order of one muon for each grain of sand on earth.*

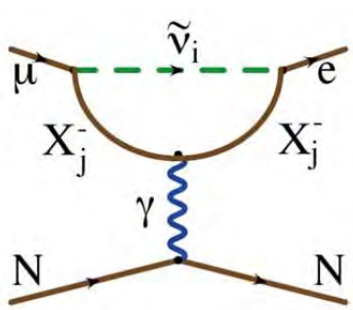


Figure 1. "Penguin" diagram for $\mu N \to eN$ decays in the Standard Model.

3. Electron cloud

An electron cloud is generated in an accelerator when stray electrons are accelerated by a passing positive beam bunch, accelerating the electrons to high enough energy that they generate secondary

* This comparison is based on the estimate of the number of grains of sand on earth from http://www.hawaii.edu/suremath/jsand.htmlhttp://www.hawaii.edu/suremath/jsand.html.

electrons when they collide with the beam pipe. Under the right conditions, a cascade effect causes the bunch to grow exponentially, eventually creating a cloud of electrons with a large enough electric field to cause beam instabilities. Electron cloud is the conceptually most complex collective effect of those we address in this work. While space charge and beam-beam interactions are calculable from first principles, electron cloud models ultimately rely on phenomenological models for the production of secondary emission of electrons. Since the details of the secondary emission yield (SEY) depend on the details of the beam pipe surface, there are unavoidably large uncertainties in the phenomenological models. We describe here two electron cloud simulation efforts. The first, in the Fermilab Main Injector, addresses the buildup of the cloud itself. It also includes a simulation of a microwave electron cloud detection experiment. The second simulation, at the CERN SPS, focuses on the destabilizing effect of a pre-generated electron cloud on the beam. It includes a simulation of a proposed compensation mechanism.

3.1. Electron Cloud Simulations in the Fermilab Main Injector

The Fermilab Main Injector is currently able to handle its designed beam intensity with losses within acceptable limits. In the Project-X era, however, the Main Injector will be required to handle approximately a factor of three increase in beam power, which could increase instabilities and lead to unacceptable losses. In the conjunction with experimental studies currently begin pursued in the Main Injector, we have performed a series of studies of electron cloud formation [4] utilizing the fully 3D and self-consistent electromagnetic PIC code VORPAL [5].

The physical model consists of an elliptical stainless steel beam pipe (of main and major axes 2.34 cm and 5.88 cm, respectively) located in a static magnetic field. Two variations were studied in detail: a short section of 0.25 m and a long section of 16 m, taken from a representative MI arc consisting of a dipole followed by a quadrupole, another dipole and, finally, a field free region. The grid used was 48×48 transversely, with 384 and 6144 longitudinal divisions for the long and short sections, respectively. In both cases, a perfectly matching layer was included at both ends of the gridded area to avoid unphysical reflections from the artificial longitudinal boundaries. The simulation included a small seed of electrons uniformly distributed along the beam direction. The details of the seed distribution proved to be irrelevant, as expected. During the simulation, there is a phase where the number of electrons grows exponentially. It was necessary to modify the ratio of macro particles to real particles during the simulation by periodically randomly culling the electron distribution. Depending of the growth speed of the cloud, roughly from 3 to 10 culling phases were needed before reaching stable saturation.

The effects on the electrons of the passage of a beam bunch can be seen in Figure 2. On the left hand side of the graph, no beam is present and the average electron kinetic energy decreases as energy is lost in collisions with the wall. When the beam enters in the center of the graph, the kinetic energy quickly increases as the electrons feel the pull of the positively-charged beam. The bunch quickly passes and the average kinetic energy again begins to decline.

Figures 3 and 4 show representative slices of the electron cloud density once saturation has been reached. Figure 3 shows transverse distributions during the relaxation (no beam present) and pinch (beam present) phases. Figure 4 shows the longitudinal distribution in the long pipe simulation. The simulation is long enough to include multiple bunches. The beam density is given by the red line.

The beam-position monitor (BPM) has been simulated via the use of VORPAL's pseudo-voltages. In a dipole, a displaced beam in an electric cloud of sufficient density ought to result in an echo of the beam pulse detectable by the BPM. An FFT analysis of the simulated BPM response shows a weak resonance at the cutoff frequency of the beam pipe. The excited electron cloud carries electromagnetic waves that can propagate in the beam pipe.

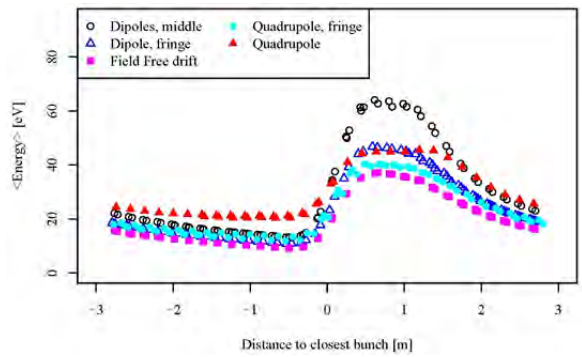
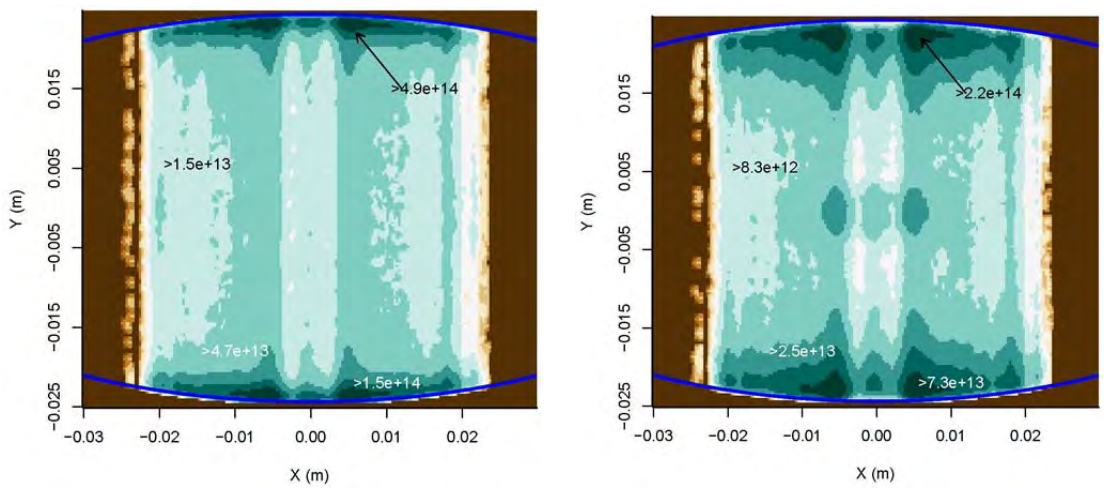


Figure 2. Variation of the kinetic energy distribution as function of the distance to the nearest bunch.



(a) (b)

Figure 3. Transverse density of the generated electron cloud in the Main Injector during the (a) relaxation and (b) pinch phases.

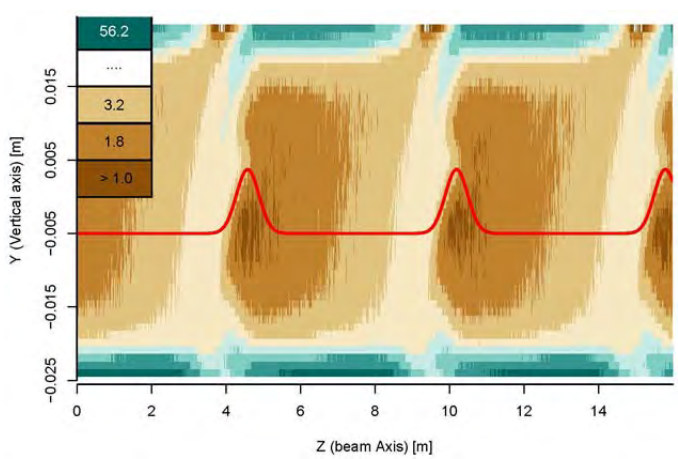


Figure 4. Y-Z projection of EC. The red line depicts the proton beam bunch profile. Here, a continuous dipole field of 0.234T is assumed.

The time scale of the electron cloud generation depends on the SEY parameters, the bunch intensity and, to a lesser extent, the magnetic field configuration. Under current conditions, in the MI dipoles, assuming a relatively high value for the maximum SEY (SEY_m) of about 2.0, the e-folding time for the generation of the cloud is about 50 ns. If SEY_m is less than 1.05, the EC dies away and saturation does not occur. Because of the uncertainties in the SEY model and the surface of the beam pipe itself, simulation alone can not *a priori* predict the EC density and its impact on the beam. However simulations provide valuable information used to guide and interpret the ongoing experimental program.

3.2. Electron Cloud in the CERN SPS

Electron clouds have been shown to trigger fast-growing instabilities on proton beams circulating in the CERN SPS [6]. A feedback system to control the instabilities is under active development [7]. To study this system, we have done a series of studies [9] of beam-electron cloud interactions and the feedback system in the SPS using WARP/Posinst [8]. Whereas the simulations in the previous section focused on the development of the electron cloud itself, this work focuses on beam-electron cloud interactions in a situation where a substantial electron cloud has already been shown to exist.

The model consists of a succession of N_s discrete interactions around the ring ("ecloud stations"). Two consecutive bunches of 1.0×10^{11} protons, representing bunches 35 and 36 of the bunch train, were modeled with an RMS bunch length of 0.23 m and an inter-bunch spacing of 25 ns. The initial electron distribution was extracted from a prior full-build-up simulation using Posinst alone to describe bunches 1 through 34, with parameters chosen to match the experimentally measured tune shift. Figure 5 shows the electron density around the two bunches projected onto the vertical and horizontal planes.

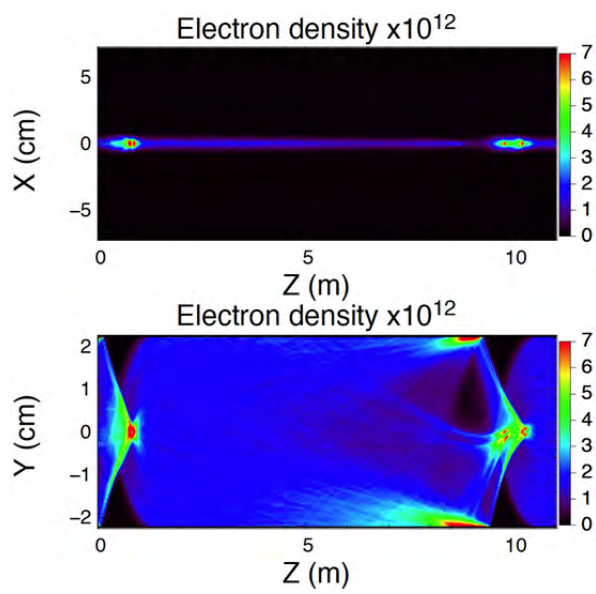


Figure 5. Electron density around two bunches in the SPS. The horizontal plane is shown on the top, the vertical on the bottom.

The feedback model takes the measured beam position at point y_{i-1} and uses the smooth focusing approximation to the full lattice to predict the beam position at time y_i ξ turns later than y_{i-1}. A gain factor g is added and a velocity kick proportional to the predicted beam position is applied. For these simulations, $\xi = 1$.

Figure 6 shows that the feedback system is successful in suppressing emittance growth. Initial comparisons with experiment show good qualitative and some quantitative agreement on key aspects of the observed instability [7]. More realistic simulations of the feedback system are planned for the future.

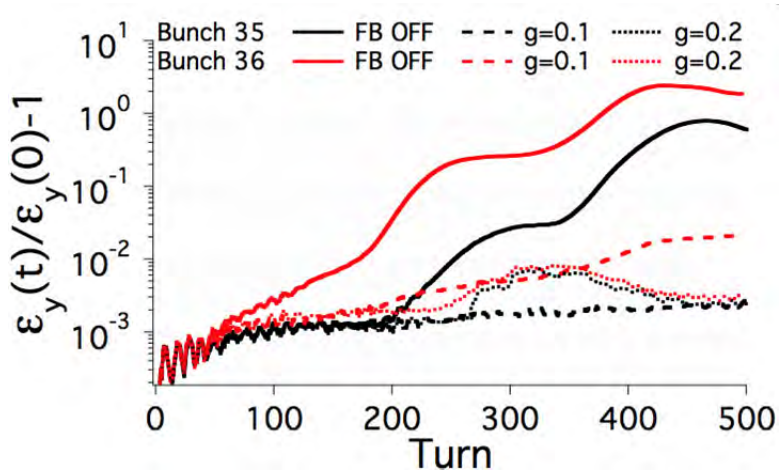


Figure 6. Emittance growth with and without compensation. g is the gain factor in the feedback system; "FB OFF" corresponds to no feedback. Note the logarithmic vertical scale.

4. Beam-beam

Parasitic interactions between colliding beams can lead to emittance growth and instabilities. They are another collective effect worthy of simulation. Beam-beam interactions may be modeled from essentially first principles, especially in comparison with electron clouds. In a typical collider beams are brought into collision, or at least near proximity, several times over the course of a turn. The basic problem is to calculate the field due to both beams during their period of close proximity and apply the effects of fields to both beams. The problem is somewhat complicated by the fact that trains of bunches are used, resulting in a combinatorial sequence of bunch i colliding first with j, then with k, etc.

The ComPASS code BeamBeam3d [10] has been developed to enable realistic simulations of beam-beam interactions. It includes an integrated, shifted Green function method for efficient field solves with spatially separated beams, a multiple-slice model for finite bunch length effects and a parallel particle-based decomposition model for ideal load balance. It can handle arbitrary closed-orbit separation, multiple bunches and multiple collision points. The optics calculations (for propagation between bunch-bunch interactions) include arbitrary transfer matrices, chromaticity effects, and thin lens multipole kicks. During the course of the ComPASS project it has been enhanced to include resistive wall wakefields, a conducting wire model, crab cavities and electron lens compensation. Figure 7 shows the model for longitudinal slices in the simulation.

BeamBeam3d has been used to model the machines at the Energy Frontier, the Tevatron and the LHC. In the case of Tevatron modeling, the combination of beam-beam interactions and chromaticity were studied [11]. The results of one of these simulations, combined with a data showing the same bunch-to-bunch behavior, are shown in Figure 8. The insight gained from these simulations was used to maximize luminosity in Fermilab's Run II. This result was described in a SciDAC Breakthrough presentation at this conference (see Reference [12]). BeamBeam3d has also been used to study beam-beam interactions at the LHC. In order consider the effects of machine optics on the beam-beam interaction, BeamBeam3d's built-in ability to do parallel parameter scans was used to survey 32 parameter combinations in parallel, effectively utilizing the availability of massively parallel computing to complete in 2.5 hours what would have taken 78 hours if jobs had to be launched individually. The results of the simulations have been fed back to the LARP team.

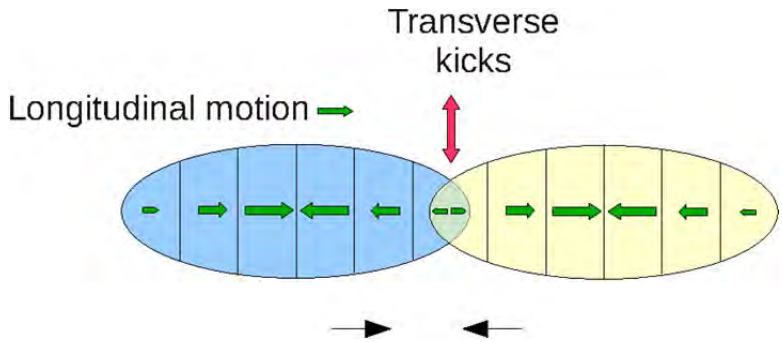


Figure 7. Longitudinal slices in the BeamBeam3d interaction calculation.

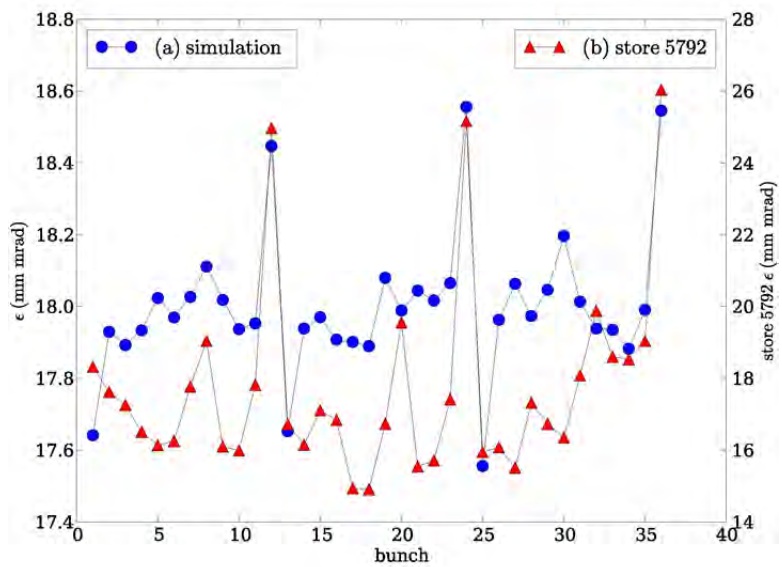


Figure 8. Vertical emittance data from the Tevatron overlaid with simulation showing similar patterns in bunch-to-bunch variation.

5. Space charge

Space charge is the canonical collective effect. It is simply the interaction of the beam with itself, and can be calculated (numerically) from very close to first principles. Space charge effects are most important for low-energy, high intensity machines. One of the ComPASS codes for studying space charge is IMPACT [13]. IMPACT contains multiple 3D Poisson solvers for a variety of boundary conditions, a detailed RF accelerating and focusing model, wakes, thin-lens nonlinear elements and more. IMPACT has recently been applied to studies for CERN's proposed PS2, the upgrade to the current CERN proton source. The results for emittance growth and beam characteristics have been used as input to the LARP design effort.

5.1. Simulations of Space Charge and Nonlinear Optics in the Fermilab Debuncher

The proposed Mu2e experiment, described above, will require running the existing Fermilab Debuncher and Accumulator rings with approximately 10^5 times the intensity seen in current operating conditions. Furthermore, the experiment plan includes resonant extraction, an inherently nonlinear process for extracting the beam from the Debuncher to the target which will create the muons for the experiment.

The combination of high-intensity beams and nonlinear optics make the Debuncher for Mu2e a prime candidate for simulation.

The ComPASS code Synergia [14] is ideal for the task. Synergia includes, among its features, fully 3D Poisson solvers for various boundary conditions and single-particle optics from CHEF [15], a highly-advanced single-particle code capable of handling all the necessary nonlinear effects at practically arbitrary levels of detail. A full simulation of the Mu2e resonant extraction process will include not only nonlinear optics and space charge, but also dynamical ramping of the lattice elements. Synergia's implementation as a fully programmable, Python-based simulation framework is perfectly suited to this problem.

Before looking at the results from Synergia simulations of Mu2e, it is necessary to introduce some basic concepts from accelerator physics, especially particle *tune*. In the presence of linear beamline elements, the motion of a particle in an accelerator is described by Hill's Equation,

$$\frac{dx^2}{d^2s} + K(s)x = 0, \tag{2}$$

where s is the distance traveled along the closed orbit and $K(s)$ is the varying quadrupole magnet strength. The solution to this equation is

$$x = \sqrt{\varepsilon\beta(s)}\cos(\phi(s) + \phi_0), \tag{3}$$

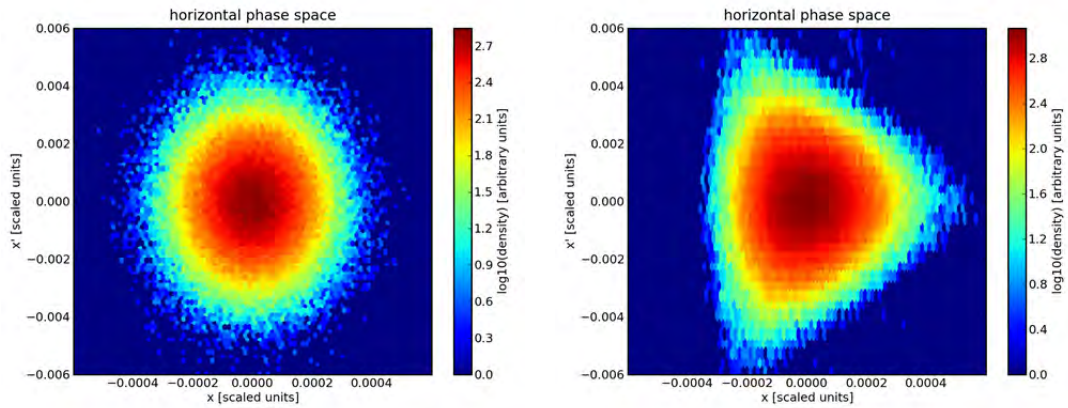
where ε and ϕ_0 are constants of the motion, $\beta(s)$ is the amplitude variation due to the varying magnet strength and $\phi(s)$ is the phase advance, which also depends on the magnet strength. The *betatron tune*, ν is defined in terms of the total phase advance in a single turn, $\Delta\phi$,

$$\nu = \frac{\Delta\phi}{2\pi} = \int_0^{s_{turn}} \frac{ds}{\beta(s)}. \tag{4}$$

Parameters used for the Synergia simulations are tabulated in Table 1. The planned cycle for the Debuncher in the Mu2e experiment is to start with a mostly linear lattice, ramp up the (nonlinear) sextupoles to create a resonance condition, then move the beam on resonance, where the it will naturally be kicked out into extraction elements. In order to tackle this problem in detail, we first consider the nonlinear problem without space charge. Figure 9 shows the horizontal phase space distribution of the initial beam in the linear lattice and the same distribution after the sextupoles have been ramped to full power. The effect is quite dramatic—it clearly shows the threefold symmetry of the sextupole field.

Table 1. Beam parameters used for Synergia simulations, taken from the original Mu2e proposal.

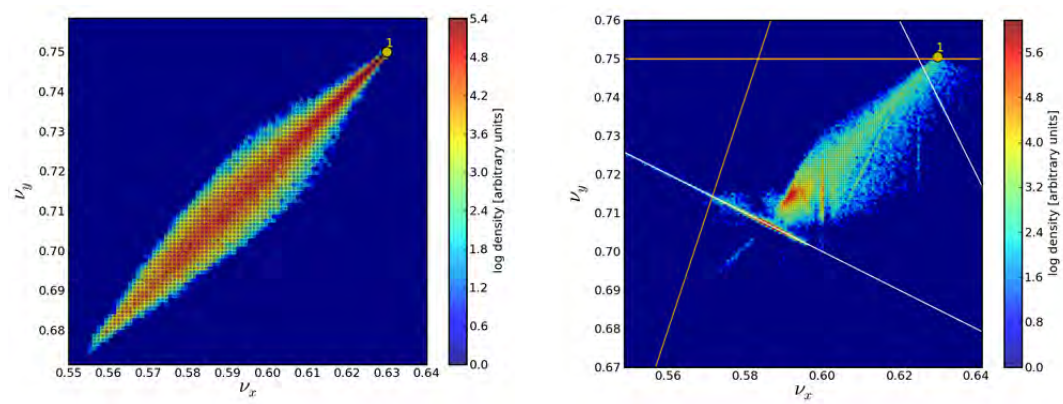
Beam energy	8 GeV
Intensity	1.2×10^{13} protons/bunch
Transverse emittance	20π mm-mr
Longitudinal width	40 nsec RMS
Space charge tune shift (Laslett)	0.08



(a) (b)

Figure 9. Horizontal phase space in the Debuncher (a) before the sextupole ramp and (b) at the completion of the sextupole ramp.

The next step is to consider the effects of space charge in the purely linear lattice. In the absence of space charge, all of the particles have the same tune, known as *bare tune*. Figure 10 shows the distribution of tunes in the beam when the space charge effect is included. The bare tune is the yellow point in the upper right. The characteristic shape of the tune footprint is known as the space-charge "necktie." Figure 10 also shows the effect of combining the nonlinear lattice with the space charge effect. The effect is striking. The system displays resonant behavior, as is evidenced by the structures that appear in the tune footprint. The system also generates enough beam loss as to be unacceptable for actual operations. The uncontrolled losses before the beam can be extracted would lead to too much radiation in the beam tunnel.



(a) (b)

Figure 10. Tune footprint including space charge and (a) linear lattice elements only, (b) linear lattice plus full-strength sextupoles. The yellow point is the bare tune.

These results can be understood in terms of resonance theory. Resonances occur when

$$m + l\nu_x + q\nu_y = \nu_x \tag{5}$$

for integer m,l,q. The smaller the integers, the lower the order of the resonance. Equation (5) implies that resonance occur along straight lines in tune space, (ν_x, ν_y). Figure 11 shows the same tune footprint, but with the 3rd-and 4th-order resonances highlighted. As one might expect, the most

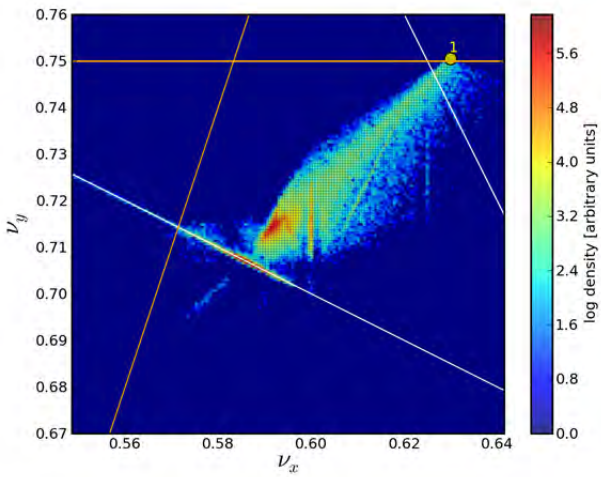


Figure 11. Tune footprint showing 3rd- and 4th-order resonance lines (white and orange, respectively). Higher order resonance structures are also present.

prominent resonance is the 3rd-order resonance, i.e., the one which is directly caused by the sextupoles. Other structures in the plot indicate the presence of higher-order resonances.

Because Synergia is capable of tracking individual particles, we can extract the trajectories of each individual particle that is lost and calculate its tune. Figure 12 dramatically demonstrates the correlation between resonance behavior and particle loss: nearly every lost particle falls neatly on a resonance line. Further structure is evident—particle loss seem to congregate around certain intersections of resonance lines. This phenomena was predicted over twenty years ago under the name "resonance streaming" [16]. It makes a stunning appearance in this analysis.

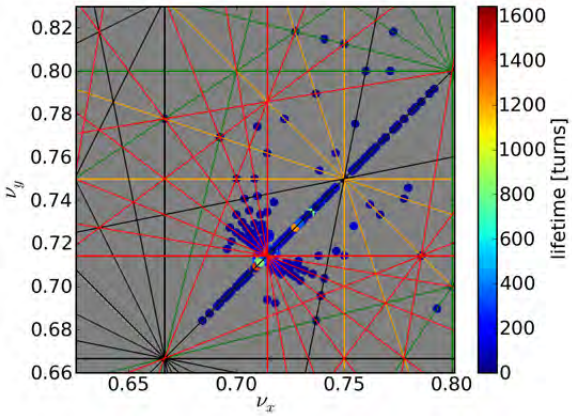


Figure 12. Lost particles in tune space and their relation to resonance lines up to 8th order: white 3rd-order, orange 4th-order, green 5th-order, black 6th-order, red 7th-order, and 8th-order. The color of the points themselves represents the lifetime of each particle before it is lost.

Because the structure of particle loss is tightly correlated with resonant behavior, it was hoped that an operational point in phase space could be found that would reduce losses to an acceptable level. After an extensive scan, a few points with somewhat better behavior were identified (along with many exhibiting much worse behavior). In the end, Synergia simulations indicate that it would be extremely difficult to run the Debuncher at this intensity with acceptable losses. This information has been fed back to the experiment, which is now considering lower-intensity operating scenarios.

6. Moving toward the Exascale

The entire HPC community is beginning to prepare for the next frontier in computing, the Exascale. ComPASS as a whole, and the beam dynamics portion of ComPASS in particular, is no exception. The first serious observation in considering the leap to the Exascale is that most, if not all, of our current parallel models are closely tied to the computing architectures available today. Indications are that

exascale computers will be very different architecturally from petascale computers. The next step is to start moving away from old assumptions by exploring the sorts of new architectures that are appearing today. Work on abstracting and parametrizing architectural assumptions in the context of generic PIC simulations have already been undertaken and shown to extract good performance from GPUs [17]. At the same time, we are exploring new decomposition strategies utilizing a full decomposition of phase space instead of just physical space with dynamical grids that track the particles, minimizing inter-processor particle movement.

A further observation about scaling beam dynamics simulations is that strong scaling is unlikely to be possible over several orders of magnitude. The types of calculations in most beam dynamics simulations simply are not large enough to scale to millions, much less billions, of processors, which also means that there are limits to straightforward weak scaling of our simulations. We are currently exploring new ways to exploit weak scaling, such as intelligent parallel optimization and new algorithms for taking longer time steps using more complex intermediate substeps.

At least one straightforward weak scaling topic is of potential interest. We currently perform macroparticle simulations of beam bunches containing $10^{10} - 10^{13}$ particles. While macroparticle calculations are usually sufficient, processes sensitive to real statistical fluctuations are difficult to model without the physical number of particles. We can easily anticipate utilizing exascale computers to simulate bunches of 10^{12} real particles.

Our next generation scaling efforts are just getting underway. We are forming collaborations between application people, computer science and applied math people and hardware vendors in order to come up with effective plans to exploit high-performance computing resources we expect to find in the upcoming era.

7. Conclusions

ComPASS beam dynamics simulations are actively involved in advancing the Energy and Intensity Frontiers in high energy physics. We have discussed how four of our codes, VORPAL, BeamBeam3d, IMPACT and Synergia have been used to perform large-scale simulations of electron cloud, beam-beam and space charge effects at CERN and Fermilab. These simulations have both benefited from and contributed to experimental and design programs in accelerator physics. We plan to extend our simulation capabilities to the next generation of high-performance computing in order to meet the needs of tomorrow's accelerators.

Acknowledgments

This work was supported by the United States Department of Energy under contract DE-AC02-07CH11359 and the ComPASS project funded through the Scientific Discovery through Advanced Computing program in the DOE Office of High Energy Physics. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research also used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

References

[1] P. Spentzouris et al. https://compass.fnal.gov/.

[2] "US Particle Physics: Scientific Opportunities A Strategic Plan for the Next Ten Years," *Report of the Particle Physics Project Prioritization Panel (P5),* http://www.er.doe.gov/hep/files/pdfs/P5_Report.

[3] Stephen D. Holmes, "Project X at fermilab," PAC'09, Vancouver, May 2009, ID-1167, http://www.JACoW.org.

[4] P. Lebrun, P. Spentzouris, J. Cary, P. Stoltz, and S. Veitzer, "Accurate Simulation of the Electron Cloud in the Fermilab Main Injector with VORPAL," IPAC'10, Japan, May 2010.

[5] C. Nieter and J. R. Cary, "VORPAL: a versatile plasma simulation code," *J. Comp. Phys.* **196**, 448–472 (2004).
[6] G. Arduini et al., CERN-2005-001, 31–47 (2005).
[7] J. D. Fox et al., SPS Ecloud Instabilities—Analysis of machine studies and implications for Ecloud Feedback, IPAC10.
[8] J-L Vay et al., "Update on e-cloud simulations using the package Warp/Posinst," PAC09, Proceedings, (2009).
[9] J-L Vay, et al., "Simulation of E-Cloud Driven Instability and its Attenuation Using a Feedback System in the CERN SPS," IPAC10.
[10] J. Qiang, M. A. Furman, and R. D. Ryne, *J. Comp.Phys.*, **198** (2004), pp. 278–294; J. Qiang, M. A. Furman, and R.D. Ryne, *Phys. Rev. ST Accel. Beams*, 104402 (2002).
[11] E.G. Stern, A. A. Valishev, J. F. Amundson, and P. Spentzouris, "Fully 3D Multiple Beam Dynamics Processes Simulations for the Tevatron," *Phys. Rev. ST AB*, 2009.
[12] E. Stern, et al., "Fully 3D multiple beam dynamics processes simulation for the Fermilab Tevatron," these proceedings.
[13] J. Qiang, R. Ryne, S. Habib, and V. Decyk, *J. Comp. Phys.* vol. 163, 434, (2000).
[14] J. Amundson, P. Spentzouris, J. Qiang, and R. Ryne, 2006, *J. Comp. Phys.* **211** 229–248.
[15] L. Michelotti, "C++ objects for beam physics," FERMILAB-CONF-91-159, *Presented at 14th IEEE Particle Accelerator Conf.*, San Francisco,CA, May 6–9, 1991.
[16] J. Tennyson, *Physica 5D* (1982) 123.
[17] V. Decyk, T. Singh, and S. Friedman, "Graphical Processing Unit-Based Simulations," *Proc. Intl. Computational Accelerator Physics Conf.* (ICAP2009), San Francisco, CA, 2009.

Wakefield computations at extreme-scale for ultra-short bunches using parallel hp-refinement

Lie-Quan Lee, Cho Ng, Arno Candel, Liling Xiao, Greg Schussman, Lixin Ge, Zenghai Li, Vineet Rawat, and Kwok Ko

SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025

E-mail: liequan@slac.stanford.edu

SciDAC CET/Institute Collaborators: E Ng, X Li, Ichitaro Yamazaki (TOPS/LBNL), D Keyes (TOPS/Columbia), Q Lu, M Shepard (ITAPS/RPI), L Dianchin (ITAPS/LLNL), K Devine, E Boman (ITAPS/CSCAPES/Sandia), K Moreland (UltaVis/Sandia), K Ma(UltaViz/UCDavis)

Abstract. The computation of wakefield induced by an ultra-short charged particle beam is one of the most computationally challenging problems in high-performance computing for accelerator. With SciDAC collaborations in computational science, a novel moving-window technique is developed for the finite element time domain (FETD) code T3P to tackle this problem through the reduction of computing needs. The window moving along with the beam in the computational domain adopts high-order finite-element basis functions through *p* refinement and/or a high-resolution mesh through *h* refinement so that a sufficient accuracy is attained with substantially reduced computational costs. Algorithms to transfer discretized fields from one mesh to another, which are the key to implementing a moving window in a finite-element unstructured mesh, are presented. Numerical experiments are carried out at DOE flagship computers using the proposed technique to compute short-range wakefields in long accelerator structures. The results are compared with those obtained from the normal FETD method and the advantages of using the moving window technique are discussed. The technique has been successfully used in evaluating wakefield effects due to ultra-short bunches for accelerator components in proposed DOE accelerators.

1. Introduction

Evaluating the effect of wakefields, the parasitic electromagnetic fields, induced by a particle beam (or bunch) transiting through an accelerator structure is one of the important tasks in accelerator design [1,2,3]. Many accelerator structures have complex geometries and large spatial dimensions compared with the size of the beam. Very often, the accurate determination of wakefields requires high-fidelity representation of geometry and large-scale parallel computation using many computer processors. For this purpose, SLAC has developed a parallel ab initio time domain tool T3P [4,5] in its finite-element computational suite ACE3P to calculate wakefields for large accelerator structures. The tool has been applied to various accelerator projects and has made tremendous impacts on accelerator design and analysis [4,5,6].

Even with the advent of parallel computation, the calculation of wakefields for a short bunch is computationally challenging. As the bunch size gets smaller, the frequency content of the wakefield excited by the bunch increases correspondingly. The mesh required for the accurate determination of

wakefields has to be fine enough to resolve the high frequency components of the bunch. For ultra-short bunches, the element size needs to be very small, and therefore the number of elements becomes very large. This makes the computation prohibitively time- consuming. When one is interested in determining the effects of wakefields excited by the bunch on itself, only the short-range wakefield around the bunch is required. In this case, only the electromagnetic fields around the bunch need to be considered and those far away can be ignored. The domain of the calculation can then be limited to a moving window around the beam as it transits through the structure. Within the moving window, accurate solution can be obtained using high-order finite element basis functions and small elements (i.e., a fine mesh). While the former increases the accuracy rapidly [7], the latter helps in resolving the high frequency content of the beam. As a result, the computational cost can be much reduced compared to when the problem is solved for the entire region of the beam transit. A moving window in structured grids is well defined and its implementation is relatively straight-forward [8,9]. This has been done in commercial electromagnetic packages for wakefield computation such as MAFIA [10]. In the finite-element analysis with unstructured meshes, a moving window cannot be easily constructed as the elements do not align in a regular pattern as in finite difference. Furthermore, the discretized field solution on the mesh inside one window cannot be readily mapped onto another. The transfer of fields from one mesh to another is non-trivial and has to be done carefully. In this article, we will address issues arising from the implementation of a moving window in an unstructured grid and present new algorithms to facilitate the calculation.

The successful application of the moving window technique to large-scale accelerator simulation would not have been possible without the collaborations with SciDAC CETs/Institutes. During SciDAC1 and the current SciDAC2 ComPASS [11] accelerator project, strong collaborations have been established in applied mathematics and computer science for advancing electromagnetic simulation using the finite-element suite ACE3P. The research and development have been focused in the areas of eigensolvers and linear solvers (TOPS), meshing and adaptive mesh refinements (ITAPS), dynamic load balancing (CSCAPES/ITAPS) and visualization (UltraVis). Detailed descriptions of the ongoing collaborations can be found in Ref. [12].

The rest of the article is organized as follows. In Section 2, we first review the finite element time domain method for electromagnetic wave propagation inside an accelerating structure. We then describe the moving window algorithms for unstructured meshes without and with mesh refinement. In Section 3, we compare the results obtained using the moving window technique with those from the whole structure simulation and verify the correctness of the algorithms. We also discuss the benefit of using the moving window technique. Finally, we provide concluding remarks in Section 4.

2. Moving Window Algorithms

2.1. Finite Element Time Domain Method

In this section, we review the formulation of the finite-element time-domain method used to solve the second-order vector wave equation obtained from Maxwell's equations. First, we present the finite element formulation of a time-domain electromagnetic boundary-value problem. Then we describe the Newmark-ß scheme used to discretize the time.

The electromagnetic fields generated by an electric current density $\vec{\mathbf{J}}$ satisfy Maxwell equations in a volume V bounded by surface S. Eliminating the magnetic field with the aid of the constitutive relations, we obtain the second-order vector wave equation, or the curl-curl equation:

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \vec{\mathbf{E}}(\vec{\mathbf{r}}, t) \right) + \varepsilon \frac{\partial^2}{\partial t^2} \vec{\mathbf{E}}(\vec{\mathbf{r}}, t) + \sigma \frac{\partial}{\partial t} \vec{\mathbf{E}}(\vec{\mathbf{r}}, t) = -\frac{\partial}{\partial t} \vec{\mathbf{J}}(\vec{\mathbf{r}}, t), \; in \quad V \tag{1}$$

where ε, μ, and σ denote, respectively, the electric permittivity, magnetic permeability, and conductivity of the medium. A set of boundary conditions may exist on different parts of the surface S, denoted by S_E, S_M, and S_R:

$$\vec{\mathbf{n}} \times \vec{\mathbf{E}}(\vec{\mathbf{r}}, t) = 0 \quad on \quad S_E \tag{2}$$

$$\vec{\mathbf{n}} \times \left(\frac{1}{\mu}\nabla \times \vec{\mathbf{E}}(\vec{\mathbf{r}}, t)\right) = 0 \quad on \quad S_M \tag{3}$$

$$\vec{\mathbf{n}} \times \left(\frac{1}{\mu}\nabla \times \vec{\mathbf{E}}(\vec{\mathbf{r}}, t)\right) + Y\vec{\mathbf{n}} \times \vec{\mathbf{n}} \times \frac{\partial}{\partial t}\vec{\mathbf{E}}(\vec{\mathbf{r}}, t) = \vec{\mathbf{U}}(\vec{\mathbf{r}}, t) \quad on \quad S_R \tag{4}$$

where $\vec{\mathbf{n}}$ represents the outward unit vector normal to S, Y denotes the surface admittance of the boundary S_R, and $\vec{\mathbf{U}}(\vec{\mathbf{r}}, t)$ is a known boundary source.

To avoid the constant, curl-free electric field that is non-physical in our simulation but is supported by Equation (1), we take the time integration of the electric field $\vec{\mathbf{E}}(\vec{\mathbf{r}}, t)$, denoted by $\vec{\varepsilon}(\vec{\mathbf{r}}, t) \equiv \int_{-\infty}^{t} \vec{\mathbf{E}}(\vec{\mathbf{r}}, \tau)\, d\tau$, and solve the following curl-curl equation instead [5]

$$\nabla \times \left(\frac{1}{\mu}\nabla \times \vec{\mathcal{E}}\right) + \varepsilon\frac{\partial^2}{\partial t^2}\vec{\mathcal{E}} + \sigma\frac{\partial}{\partial t}\vec{\mathcal{E}} = -\vec{\mathbf{J}}, \tag{5}$$

We discretize the field using tangentially continuous Nédélec basis functions [14], $\vec{\varepsilon}(\vec{\mathbf{r}}, t) = \sum_{i=1}^{N} x_i(t)\, \vec{\mathbf{N}}_i(\vec{\mathbf{r}})$, with N denoting the total number of unknowns, and obtain the equation in the matrix and vector form:

$$\mathbf{K}\mathbf{x} + \mathbf{M}\frac{\partial^2}{\partial t^2}\mathbf{x} + (\vec{\mathbf{r}} + \mathbf{Q})\frac{\partial}{\partial t}\mathbf{x} = \mathbf{f} \tag{6}$$

where $\mathrm{x} \equiv (x_1, x_2, ..., x_N)^T$ and $\mathbf{K}$, $\mathbf{M}$, $\vec{\mathbf{r}}$, $\mathbf{Q}$ are N-by-N square matrices given by

$$\mathbf{K}_{ij} = \int_V \frac{1}{\mu}\left(\nabla \times \vec{\mathbf{N}}_i(\vec{\mathbf{r}})\right) \cdot \left(\nabla \times \vec{\mathbf{N}}_j(\vec{\mathbf{r}})\right) \mathrm{d}V \tag{7}$$

$$\mathbf{M}_{ij} = \int_V \varepsilon\vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \cdot \vec{\mathbf{N}}_j(\vec{\mathbf{r}})\, \mathrm{d}V \tag{8}$$

$$\vec{\mathbf{r}}_{ij} = \int_V \sigma\vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \cdot \vec{\mathbf{N}}_j(\vec{\mathbf{r}})\, \mathrm{d}V \tag{9}$$

$$\mathbf{Q}_{ij} = \int_{S_R} Y\left(\vec{\mathbf{n}} \times \vec{\mathbf{N}}_i(\vec{\mathbf{r}})\right) \cdot \left(\vec{\mathbf{n}} \times \vec{\mathbf{N}}_j(\vec{\mathbf{r}})\right) \mathrm{d}S \tag{10}$$

The vector $\mathbf{f}$ is the driving force that is given by

$$\mathbf{f}_i = -\int_V \vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \cdot \vec{\mathbf{J}}(\vec{\mathbf{r}}, t)\, \mathrm{d}V - \int_S \vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \cdot \left(\int_{-\infty}^{t} \vec{\mathbf{U}}(\vec{\mathbf{r}}, \tau)\, \mathrm{d}\tau\right) \mathrm{d}S \tag{11}$$

In solving the above equation, the implicit Newmark-ß scheme [15] is employed for numerical time integration. The method is proven to be unconditionally stable when the prescribed parameter ß is larger than or equals to 0.25. With the Newmark-ß time discretization, a system of linear equations is needed to be solved at each time step:

$$\left[\mathbf{M} + \tfrac{c\Delta t}{2}(\vec{\mathbf{r}} + \mathbf{Q}) + \beta(c\Delta t)^2\mathbf{K}\right]\mathbf{x}^{n+1} = \mathbf{b} \tag{12}$$

$$\begin{aligned} with \quad \mathbf{b} \equiv \; & \left[2\mathbf{M} - (1-2\beta)(c\Delta t)^2\mathbf{K}\right]\mathbf{x}^n \\ & - \left[\mathbf{M} - \tfrac{1}{2}c\Delta t(\vec{\mathbf{r}} + \mathbf{Q}) + \beta(c\Delta t)^2\mathbf{K}\right]\mathbf{x}^{n-1} \\ & -(c\Delta t)^2\left[\beta\mathbf{f}^{n+1} + (1-2\beta)\mathbf{f}^n + \beta\mathbf{f}^{n-1})\right] \end{aligned}$$

In the above equation, vectors with superscripts $n + 1$, n, and $n - 1$ represent the ones at the current and the two previous steps, respectively. At a given discrete time t, the electric field $\vec{\mathbf{E}}$ and the magnetic flux density $\vec{\mathbf{B}}$ can be computed from the solution vector $\mathbf{x}$:

$$\vec{\mathrm{E}}(\vec{\mathrm{r}}) = \sum_i \frac{\partial}{\partial t} x_i \vec{\mathrm{N}}_i(\vec{\mathrm{r}}) \tag{13}$$

$$\vec{\mathrm{B}}(\vec{\mathrm{r}}) = -\sum_i x_i \nabla \times \vec{\mathrm{N}}_i(\vec{\mathrm{r}}) \tag{14}$$

Once the time histories of the electric field and the magnetic flux density have been determined, observables depending on them can be evaluated. For example, a wake potential generated by a moving particle beam with charge q inside an electromagnetic structure and seen by a test particle traveling on the same or on a parallel path at a distance s behind the particle beam can be calculated as follows:

$$W(\vec{\rho}', \vec{\rho}, s) = -\frac{1}{q} \int_{z_1}^{z_2} E_t(\vec{\rho}, z, t)|_{t=\frac{z+s}{c}} \, \mathrm{d}z \tag{15}$$

Here, without loss of generality, the transit direction of the beam through the structure and that of the test charge is in the positive z direction. c is the speed of light. The transverse offsets of the driving beam and the test charge from the z axis are $\vec{\rho}'$ and $\vec{\rho}$, respectively. The test particle enters the structure at z_1 and leaves at z_2.

2.2. Moving Window Technique using hp-Refinement on Finite Element Mesh

In calculating the short-range wakefield excited by a moving particle beam inside a large and long accelerator structure, only the small region in the vicinity of the particle beam is required in the simulation. The moving window technique, in which the domain of the simulation is limited to a small region of interest near the beam and moves along with it, can greatly reduce computational resource requirements. This technique has been applied to finite difference time domain (FDTD) methods [8,9], but not to finite element based methods on unstructured grids. In this section, we describe the algorithm for the implementation of a moving window for the finite element time domain simulation method on unstructured grids. More detailed discussion of the algorithm can be found in reference [13].

Given an unstructured grid that represents the discretized geometry and a moving particle beam, first we define a bounding box containing the beam as the window. The transverse dimension of the bounding box is large enough to include that of the mesh while the longitudinal dimension covers the beam size and the padded zones in front of and behind the beam, as illustrated in Figure 1. The computational domain contains all the mesh elements inside the window. A mesh element is inside the window when any part of the element overlaps with the bounding box. Each element inside the window can be optionally subdivided (uniform h-refinement) if the higher-resolution is required to resolve a short beam. Its corresponding basis functions order p is set to be nonzero. Otherwise, the mesh element is outside the window and excluded from the computation by setting its corresponding basis functions order p to zero.

The length of the padded zone behind the beam, b, is determined by the maximum distance s_{max} of the wakefield that needs to be computed. Namely, $b + d >= s_{max}$, where d is the longitudinal size of the driving bunch. The length of the padded zone in front of the beam, f, is determined by how often the computational domain (the volume of the moving window) changes. The shorter the f is, the smaller the computational domain is and the more frequent the window moves. Each time when the window moves, there are additional computational costs in establishing the mesh inside the new window, partitioning the mesh and distributing the mesh elements for load balancing if the simulation runs in

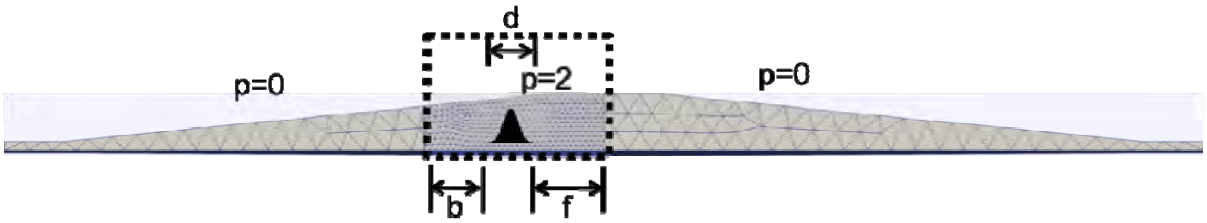
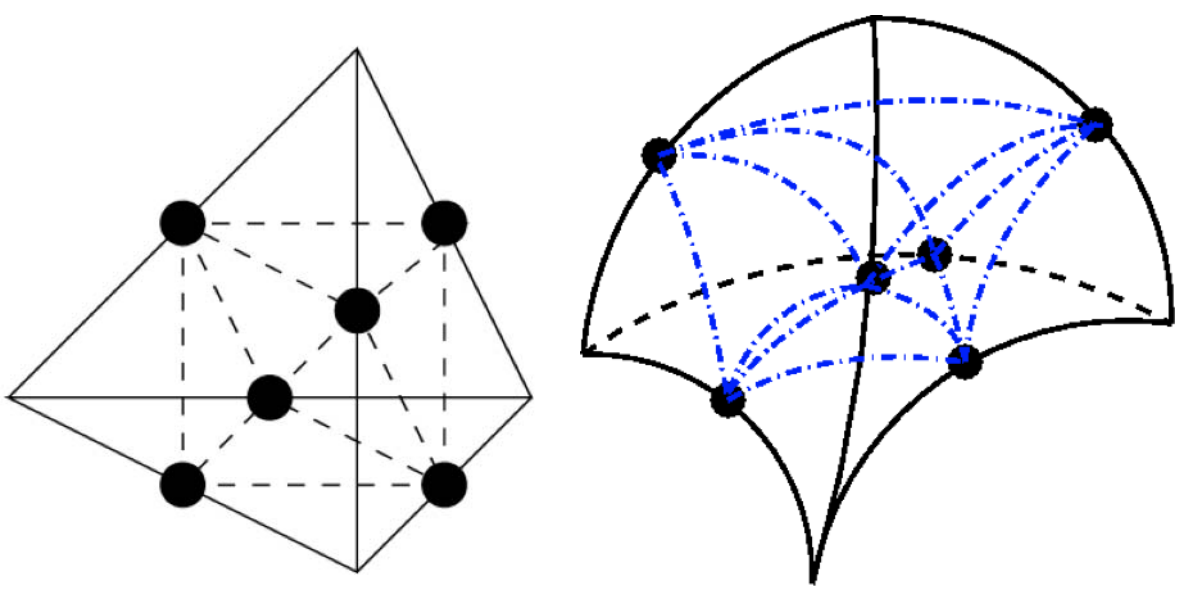


Figure 1. Moving window on an unstructured grid. The dashed box illustrates the moving window region. Its length is defined by $b + d + f$, where b is the length of the padded zone behind the beam, d the beam size, and f the length of the padded zone in front of the beam. Initial b and f are adjustable parameters in each specific problem. The particle beam moves from left to right.

parallel, assembling the matrices in Equations (7–10), and transferring vectors from the old mesh to the new mesh. The optimal choice of f would be to balance the reduction of the computational domain using a smaller window and the additional overheads of switching to a new window. However, a priori method for the determination of the optimal f does not exist. In practice, we choose f so that the beam will march forward for about a hundred time steps before it reaches the front (right) boundary of the window. In the unstructured grid, generally speaking, the left and right boundaries of the mesh in the window do not align smoothly with the planar surfaces of the bounding box. This will not affect the computation accuracy as long as the resultant computational domain covers the wakefield distance to be calculated.

Curvilinear tetrahedral elements are used in our simulation for high-fidelity representation of geometry. In refining an existing coarse mesh, we choose to subdivide a tetrahedron by splitting all the six edges into shorter ones [16]. Cutting a tetrahedron at its four corners leaves an octahedron, which can be split into four tetrahedra by adding an inner edge connecting two diagonally opposite corners of the octahedron. There are three possible inner diagonals in the octahedron, and the shortest one is chosen to minimize the distortions of the created tetrahedra. Figure 2 illustrates the edge splitting in subdividing a linear or curvilinear tetrahedron.



(a) Subdividing a linear tetrahedron. (b) Subdividing a curvilinear tetrahedron.

Figure 2. Subdividing tetrahedral elements by splitting all the six edges.

In the rest of the section we will discuss how to move the window. When the particle beam reaches the right boundary of the window, the bounding box of the window is moved along the beam transit direction so that it has a padded zone with length f in front of the beam and a padded zone with length b behind, as shown in Figure 3. The computational domain is updated by adding new elements entering the new window and by dropping elements in the old window when they are outside the new window. There is a shared mesh region between the old and new meshes, as illustrated in Figure 3. The new mesh is refined with the method discussed above and partitioned using ParMetis [17]. The elements in the mesh are redistributed to corresponding processes. Then, the matrices in Equations (7–10) are re-assembled.

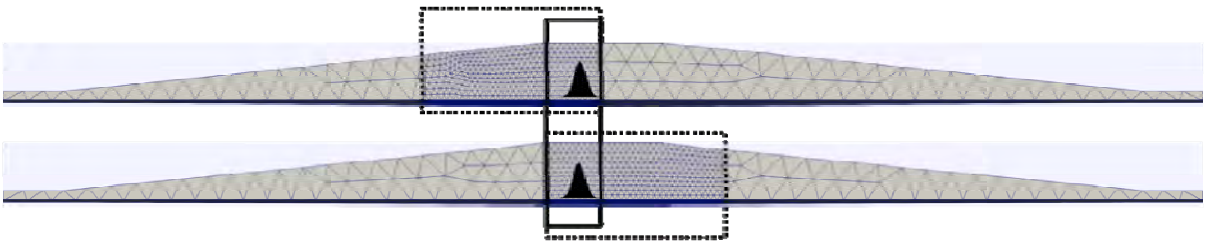


Figure 3. The window is moved along the beam transit direction when the beam front reaches the right boundary of the window. The dashed boxes illustrate the moving window regions. The top and bottom plots represent two consecutive windows. The solid box is the overlapping computational domain between the old and new meshes.

In order to continue time marching as in Equation (12), the vectors $\mathbf{x}^n$, $\mathbf{x}^{n-1}$, $\mathbf{f}^n$, and $\mathbf{f}^{n-1}$ that are defined on the old mesh need to be transferred a new set of vectors defined on the new mesh.

A novel method [18] has been recently developed to project discretized electromagnetic fields from one mesh onto another. The details can be found in Reference [18]. Here we only briefly review the method.

Given a discretized vector field $\mathbf{e}^{old}$ with a set of basis functions $\{\vec{\mathbf{N}}_i^{old}\}$ on an unstructured mesh, we need to obtain the discretized field $\mathbf{e}^{new}$ with a different set of basis functions $\{\vec{\mathbf{N}}_i\}$ on a new mesh that accurately represents the same field $\mathbf{e}^{old}$ on the old mesh. Algorithm 1 described the details of the method. Note that α is an adjustable parameter to balance the errors of the field and its curl, which are introduced by the projection of the field from one mesh to another. If α is set to zero, the above method is equivalent to interpolation using a different set of vector basis functions. However, the curl of the field will contain very large errors. It is 1 recommended to use $\alpha = \frac{1}{4}(c\Delta t)^2$ in the simulation where c is the speed of light and Δt the 4 time step.

3. Results and Discussions

In this section, we present an example to demonstrate the correctness and effectiveness of using the moving window technique in the FETD method for wakefield calculations in accelerator applications. We then present two results applying the moving window technique with hp refinement. One for the wakefield of 0.6 mm bunch inside a vacuum chamber in the Energy Recovery Linac. And the other is the short-range wakefield of a 0.3 mm bunch inside the coupler of a cavity designed for International Linear Collider [19]. More detailed discussion about verification and validation can be found in reference [13].

The example is to calculate the short-range wakefield of a wiggler taper structure in PEP-X [1], currently under study as a new generation synchrotron light source. The geometry of the wiggler taper is shown in Figure 4. It provides smooth transitions from a 45 mm × 15 mm rectangular pipe to 48 mm circular beam pipe and back to a 45 mm×15 mm rectangular pipe. The length of each transition is

Algorithm 1 Transfer a vector from the old mesh to the new mesh.

input: given a vector $\mathbf{e}^{old}$ on the old mesh
output: a new vector $\mathbf{e}^{new}$ on the new mesh

(i) define a field $\vec{\mathbf{F}}(\mathbf{e}^{old}) \equiv \sum_i e_i^{old} \vec{\mathbf{N}}_i^{old}$

(ii) define the curl of the field $\nabla \times \vec{\mathbf{F}}(\mathbf{e}^{old}) = \sum_i e_i^{old} \nabla \times \vec{\mathbf{N}}_i^{old}$

(iii) evaluate discretized vector $\mathbf{b}$ on the new mesh
$\mathbf{b}_i \equiv \int \vec{\mathbf{F}}(\mathbf{e}^{old}) \cdot \vec{\mathbf{N}}_i + \alpha \left[\nabla \times \vec{\mathbf{F}}(\mathbf{e}^{old})\right] \cdot (\nabla \times \vec{\mathbf{N}}_i)\, \mathrm{d}V$
where α is an adjustable parameter

(iv) solve the system of linear equations to obtain $\mathbf{e}^{new}$
$(\mathbf{M} + \alpha \mathbf{K})\mathbf{e}^{new} = \mathbf{b}$

Figure 4. The side view of a computer model of the wiggler taper in PEP-X. It provides smooth transitions from a 45 mm × 15 mm rectangular pipe to 48 mm circular beam pipe and back to a 45 mm × 15 mm rectangular pipe. The transition length is 400 mm each. The circular beam pipe is 100 mm long and the rectangular beam pipes are both 50 mm long. The total length of the model is 1 m.

400 mm. The circular beam pipe is 100 mm long and the rectangular beam pipes are both 50 mm long. The total length of the model is 1 m. We need to evaluate the wakefield due to a short particle beam passing through the structure.

We did the following computational experiment to verify our moving window with the mesh refinement algorithm described in the last section. First, we generated a mesh with 5 mm element size for the simulation using a beam with σ = 20 mm bunch. Then, we generated a mesh with 10 mm element size. For this coarser mesh, we ran the simulation using a moving window with one level of mesh refinement for the same beam size. We used a window size of 0.5 m with b = 0.1 m and f = 0.2 m. Both runs were carried out using 5 nodes of a Cray XT5 computer with each node containing two hex-core AMD Opteron processors, 16GB memory, and a SeaStar 2+ route. The wakefields from both runs are shown in Figure 5, and the results are in remarkable agreement for the first 0.3 m. That verifies the correctness of our algorithm for the moving window with mesh refinement. The technique of moving window with mesh refinement further reduces the computational cost, especially in terms of memory usage since only the refined mesh inside the window needs to be stored.

With the moving window FETD method, we calculated the wake potential for a short particle beam with σ = 0.6 mm in a vacuum chamber in the Energy Recovery Linac. The geometry is shown in Figure 6a and the wake plotted in Figure 6b. The initial coarse mesh was constructed using 0.6 mm element size and mesh elements inside the moving window with b = 1 mm and f = 12 mm was uniformly refined to 0.3 mm element size. It took about 5 hours with 18,000 cores of Jaguarpf, a Cray XT5 to complete the simulation. Figure 7 shows a snapshot of electromagnetic field distribution inside a vacuum chamber in the Energy Recovery Linac in calculating the wake potential with a moving window technique using hp-refinement.

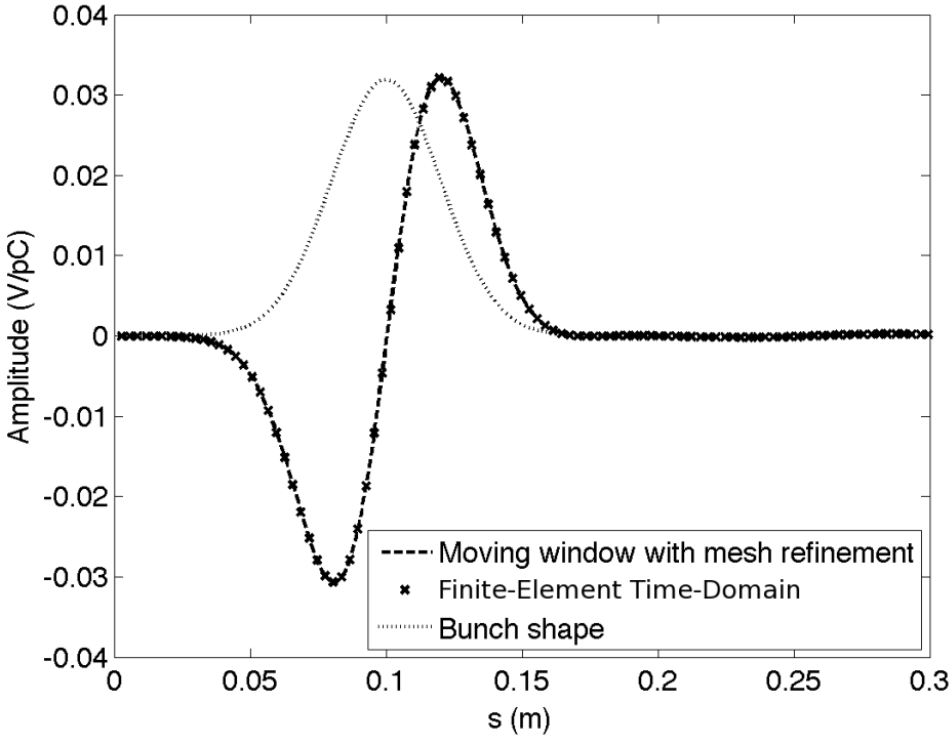
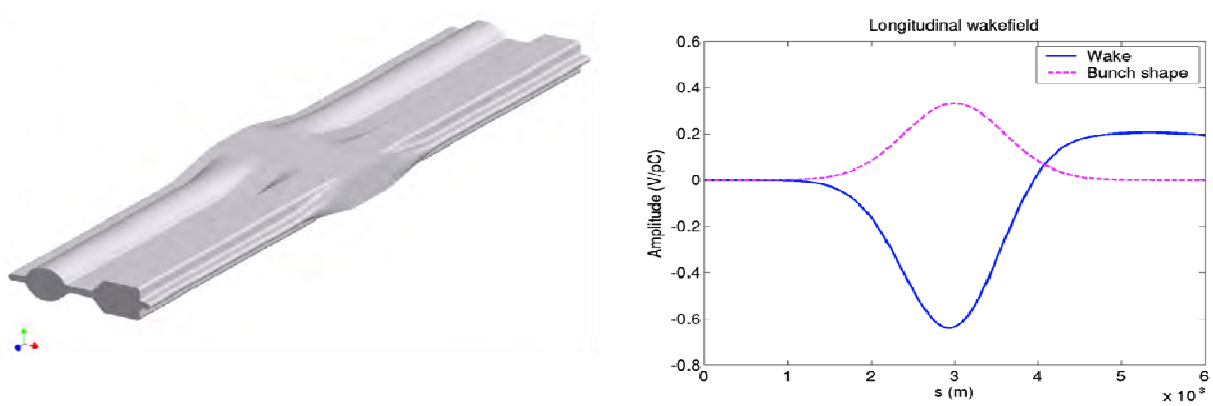


Figure 5. Comparison of wake potential due to a particle beam with σ = 20 mm calculated from two runs. The first run uses a moving window with mesh refinement. The window size is 0.5 m in both cases with b = 0.1 m and f = 0.2 m. The initial mesh size in the first run is 10 mm. The second run use traditional FETD approach with a mesh size of 5 mm.



(a) (b)

Figure 6. Wake potential of a σ = 0.6 mm bunch passing through a vacuum chamber in the Energy Recovery Linac, calculated with a moving window technique using hp-refinement. (a) A computer model of the vacuum chamber. (b) Wake potential.

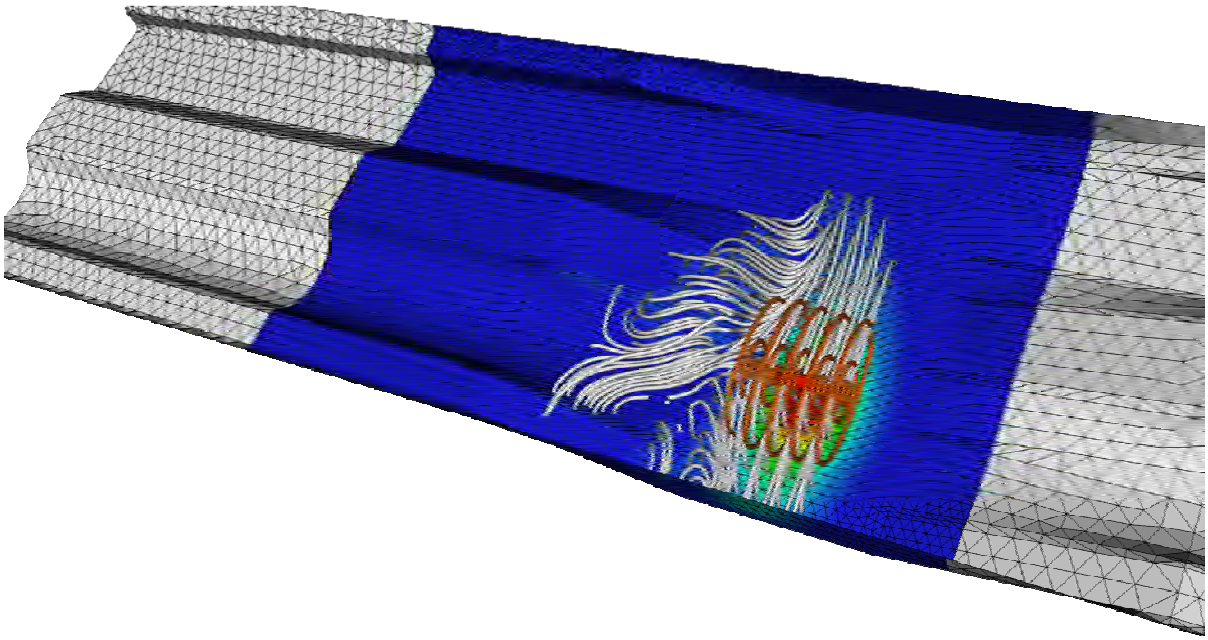
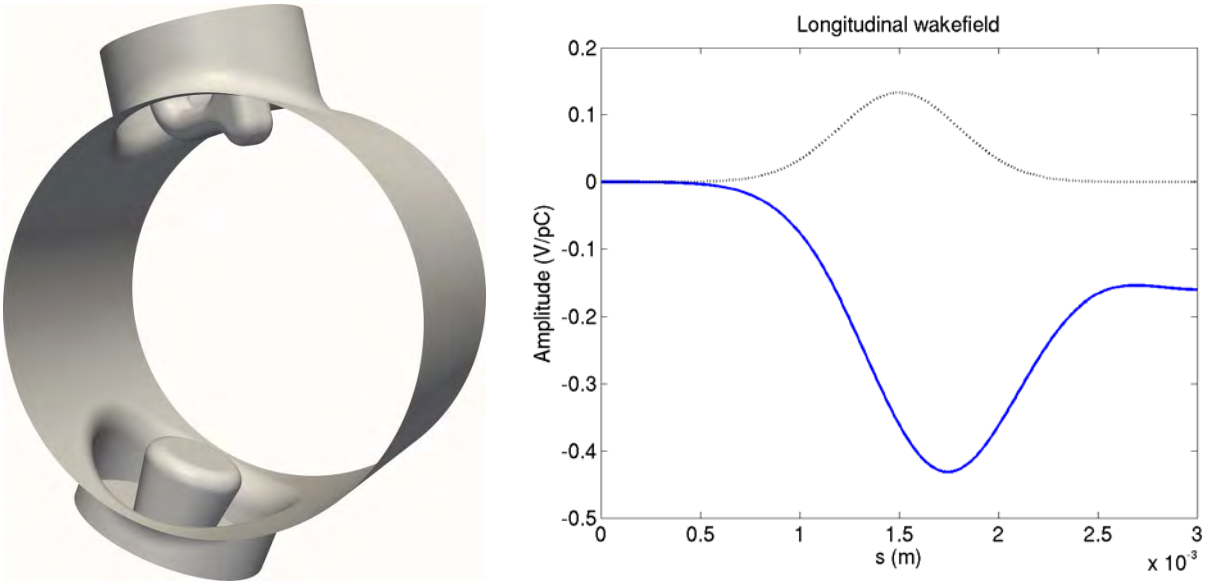


Figure 7. A snapshot of electromagnetic field distribution inside a vacuum chamber in the Energy Recovery Linac in calculating wake potential with a moving window technique using hp-refinement. An oversized red pill represents a bunch moving particles. On the surface pseudo-coloring shows the electric field strength using colors of the rainbow, where low through high values correspond to, blue through red. The regions outside the moving window are colored grey and are not included in the computation. Electric field lines are shown in white. Magnetic field lines form loops and are shown in orange.

Finally, we present a preliminary result of wakefield of a 0.3 mm bunch inside a coupler in the accelerating cavity designed for the International Linear Collider. The geometry of the coupler is shown in Figure 8a. We used mesh element size of 0.15 mm in the simulation with the moving window technique. There are about 30 million elements inside the moving window with b = 0.6 mm and f = 3 mm. We used the first-order basis functions in this simulation. The calculated wake potential plotted in Figure 8b. It took about 8 hours with 12,000 cores of Jaguarpf, a Cray XT5 to complete the simulation. We are carrying out the simulation with the 2nd order basis functions to provide more accurate wakefield result in the ILC coupler.

4. Concluding Remarks

It is computationally challenging to calculate wakefields induced by a ultra-short particle beam. With SciDAC collaborations, a novel moving-window technique for the finite element time domain (FETD) method is developed to reduce the computing needs. The window moving along with the beam in the computational domain adopts high-order finite-element basis functions through p refinement and/or a high-resolution mesh through h refinement so that a sufficient accuracy is attained with substantially reduced computational costs. Algorithms to transfer discretized fields from one mesh to another, which are the key to implementing a moving window in a finite-element unstructured mesh, are presented. Numerical experiments are carried out at DOE flagship computers using the proposed technique to compute short-range wakefields in long accelerator structures. The results are compared with those obtained from the normal FETD method and the advantages of using the moving window technique are discussed. The techniques have been successfully used in designing accelerator components for proposed DOE accelerators.



(a) (b)

Figure 8. Wake potential of a σ = 0.3 mm bunch passing through a coupler in the accelerating cavity designed for the International Linear Collider. (a) The computer model of the coupler region. (b) The grey dotted line presents the bunch shape while the blue line is the longitudinal wakefield.

Acknowledgments
The authors would like to thank Andreas Kabel and Mark Shephard for insightful discussions. This work was supported by the U.S. Department of Energy under contract number DE-AC02-76SF00515. Part of the work used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory and the National Energy Research Scientific Computing Center, which are supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725 and DE-AC02-05CH11231, respectively.

References

[1] Bane K et al. 2010. A design report of the baseline for PEP-X: an ultra-low emittance storage ring Tech. rep. SLAC National Accelerator Laboratory.
[2] Zennaro R. 2007. Considerations on short range wakefield for CLIC main beam accelerating structures Tech. rep. EUROTeV eUROTeV-Report-2007-072.
[3] Hoffstaetter G H. 2005. Toward the energy recovery linac, a brighter x-ray. Source URL http://erl.chess.cornell.edu/.
[4] Ko K et al. 2006. SciDAC and the international linear collider proc. of the 2006 SciDAC Conference.
[5] Candel A et al. 2006. Parallel higher-order finite element method for accurate field computations in wakefield and pic simulations. Proc. of the 9th International Computational Accelerator Physics Conference. URL http://icap2006.web.cern.ch/icap2006/.
[6] Candel A, Kabel A, Lee L Q, Li Z, Ng C, Schussman G, and Ko K. 2009. *Journal of Physics Conference Series*, **180** 012004–012013.
[7] Ainsworth M. 2004. *Math. phys. eng. sci.* **362** 471–492.
[8] Fidel B, Heyman E, Kastner ,R and Ziolkowski R W. 1997. *Journal of Computational Physics.* **138** 480–500 ISSN 0021-9991.
[9] Luebbers R, Schuster J, and Wu K. 2003. *Military Communications Conference, 2003. MILCOM 2003. IEEE.* vol 2 pp. 1397–1401.

[10] 2010 MAFIA a multi-purpose system designed to solve electromagnetic problems computer Simulation Technology. URL http://www.cst.com/Content/Products/MAFIA/Overview.aspx.
[11] The (ComPASS) SciDAC accelerator homepage http://compass.fnal.gov/.
[12] Lee L Q. 2009. Computational science R&D for electromagnetic modeling: Recent advances and perspective to extreme-scale. ComPASS Colaborative Meeting 2009.
[13] Lee L Q, Candel A, Ng C, and Ko K. Submitted *Journal of Computational Physics*.
[14] Sun D K, Lee J F, and Cendes Z. 2001. *SIAM J. SCI. COMPUT*, **23** 1053–1076.
[15] Newmark N M. 1959. *Journal of Engineering Mechanics Division*, **85** 67–94.
[16] Ruprecht D and Muller H. 1998. *Mathematical Visualization*, ed Hege H C and Polthier K (Heidelberg: Springer Verlag), pp. 61–70. URL http://citeseer.ist.psu.edu/223528.html.
[17] Schloegel K, Karypis G, and Kumar V. 2002. *Concurrency and Computation: Practice and Experience*, **14** 219–240.
[18] Lee L Q, Candel A, Kabel A, and Li Z. 2008. On projecting discretized electromagnetic fields with unstructured grids. Tech. rep. SLAC National Accelerator Laboratory. URL http://www.slac.stanford.edu/pubs/slacpubs/13250/slac-pub-13333.pdf.
[19] The international linear collider (ILC) homepage http://www.linearcollider.org/.

Dream beams: Extreme-scale computing enabling new accelerator technologies for the energy and intensity frontiers

W.B. Mori,[1] W. An,[1] V.K. Decyk,[1] W. Lu,[1] F.S. Tsung,[1] R.A. Fonseca,[2] S.F. Martins,[2] J. Vieira,[2] L.O. Silva,[2] M. Chen,[3] E. Esarey,[3] C.G.R. Geddes,[3] W.P. Leemans,[3] C.B. Schroeder,[3] J.-L. Vay,[3] K. Amyx,[4] D.L. Bruhwiler,[4] J.R. Cary,[4] E. Cormier-Michel,[4] B. Cowan,[4] P. Messmer,[4] C. Huang,[5] and T. Antonsen[6]

[1] University of California, Los Angeles, CA 90095, USA
[2] Instituto Superior Tecnico, Lisbon 1049-001, Portugal
[3] Lawrence Berkeley National Lab, Berkeley, CA 94720, USA,
[4] Tech-X Corporation, Boulder, CO 80303, USA
[5] Los Alamos National Laboratory, NM 87545, USA
[6] University of Maryland, College Park, MD 20742, USA

Email: mori@physics.ucla.edu

Abstract: For the past 80 years, the instrument of choice in experimental high-energy physics has been particle accelerators, with the newest being the Large Hadron Collider (LHC) at CERN. The construction cost alone for the LHC machine is nearly 10 billion dollars. It is clear that if the same technology is used then the world's next atom smasher" at the energy frontier will cost at least several time that in today's dollars, making it unlikely to be built. One might ask what to do next? One way is to improve the intensity of the particle beams to be collided against each other and keep their energy constant, while another is to develop new accelerator technologies that could reduce the size and cost of a future higher energy machine. In this article we discuss how large scale simulations using state-of-art codes have played and will continue to play a central role in the development of plasma-based acceleration which is currently the leading candidate for a "new" accelerator technology. We will describe the current status of algorithms and codes including recent developments, how these codes have led to scientific discovery, how they were used to approve to new experimental facilities (FACET and BELLA), how they are being used to develop collider concepts based on plasma-based acceleration, and what advances in simulation capability can be expected in the near future.

1. Introduction

Particle accelerators are critical to scientific discovery in the United States Department of Energy (DOE) program and indeed the world. Of the 28 facilities listed in the 2003 DOE report Facilities for the Future of Science: A Twenty-Year Outlook, 14 involve accelerators [Abraham03]. The development and optimization of accelerators are essential for advancing our understanding of the fundamental properties of matter, energy, space, and time, and for enabling research in aspects of materials science, chemistry, geosciences, and bioscience. Modeling of accelerator components and simulation of beam dynamics are necessary for understanding and optimizing the performance of existing accelerators and for optimizing the design and cost effectiveness of future accelerators.

In the next decade, the High Energy Physics (HEP) community will explore the energy frontier by operating and upgrading the Large Hadron Collider (LHC), by developing novel concepts and technologies necessary for the design of the next lepton collider, and through newly approved operating facilities such as BELLA and FACET at which R&D for these new acceleration technologies will be undertaken. It will also be exploring the intensity frontier by designing high intensity proton sources for neutrino physics and rare process program uses accelerators to study the properties of nuclear matter and the structure of the nucleus, understand the mechanism of quark confinement, and create and study the quark-gluon plasma state. The flagship DOE Nuclear Physics (NP) accelerators are CEBAF at Jefferson Lab and RHIC at Brookhaven National Laboratory (BNL). The worldwide nuclear physics community has identified the construction of a polarized electron-ion collider as a long-term objective in the ongoing effort to understand and answer questions in these areas. DOE/NP is considering two approaches for such a facility: e-RHIC and ELIC. DOE/NP is also constructing a facility for rare isotope beams (FRIB) that will permit studies of nuclei far from stability that promise to radically improve our understanding of atomic nuclei.

Today's accelerators are complex, large, and expensive. For example, the LHC is 30 kilometers in circumference and cost over 10 billion dollars to build. Designing and developing a new accelerator or upgrading an existing one is not only time consuming but is also expensive. Computer simulations have the potential to greatly reduce the time and cost it takes from the conceptual design of an accelerator to operation.

In this article, we will concentrate on the role that simulations have played and will continue to play in the development of a new accelerator technology for a lepton collider at the energy frontier. This technology is plasma-based acceleration [Joshi06, Esarey96]. In plasma based acceleration a short and intense laser or relativistic particle beam (the driver) propagates through a plasma near the speed of light. The light pressure of the laser or the space charge forces from the particle beam displaces plasma electrons. The ions then pull the electrons back towards where they started thereby creating a plasma wave wake with a phase velocity near the speed of light. Schematics of both are shown in Figure 1. The accelerating (electric) field in these wakes are more than 1000 times higher than those in existing accelerators. Properly shaped and phased electrons or positron beams (witness beams) are loaded onto the wake and they surf to ultra-high energies in very short distances.

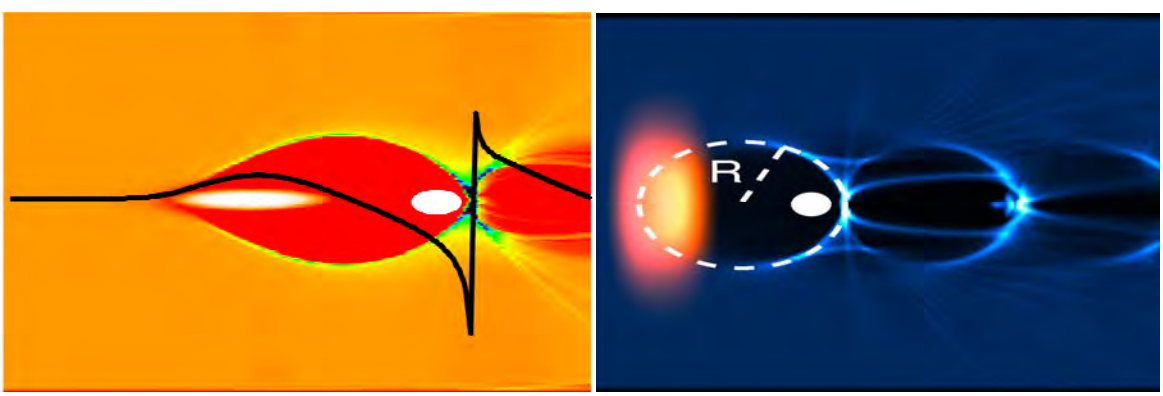


Figure 1. On the left is an electron beam (white) moving from right to left. It forms a wakefield (density of plasma is shown. A lineout of the accelerating field is shown in black. A trailing bunch is shown in white in the back of the wakefield. On the right a laser (orange) is moving from right to left. It also creates a wakefield. The wakefield in both cases is a moving bubble of a radius R. A trailing beam is shown in white as well.

2. Hierarchy of Algorithms and Codes

From the very beginning computer simulation has played a central role in the advancement of plasma based acceleration. The very first paper on the subject (it was using a laser as the driver) [Tajima79] was completely based on simulations (one dimensional), as the laser technology did not exist (in fact one could argue that the proper type of laser did not exist until a few years ago). Since that time, simulations have moved from being used to test new ideas, to modeling the full scale of experiments in three dimensions, to being used to design new facilities, and to being used to test key parts of a collider concept for parameters not yet accessible in a laboratory. As in most fields this was achieved through a combination of improved and more sophisticated code, parallelization of code, new algorithms including reduced models (that were developed through an improved understanding of the theory), and faster and larger computers. Much of this work was done under the SciDAC program including the current COMPASS project. As we will argue later, new hardware and new software developed for the hardware, together with improved algorithms, and workflows will lead to simulations being used for real-time steering of experiments, being used with genetic algorithms to find optimal laser and particle beam profiles etc, as well as simulations which integrate plasma stages with other components in a complex accelerator.

To accurately simulate this process one needs to model how a short and intense driver evolves over "large" distances, how the wake is excited and how it evolves, and how the properties of the witness beams evolve as they are accelerated. In many cases the excitation of the wake is nonlinear so a particle or kinetic description is needed. The leading kinetic description is the particle-in-cell (PIC) method. In some cases fluid models for the plasma response may be useful, however PIC algorithms are now getting very fast and reduced PIC models can have very little noise. It is worth noting that currently ~12% of INCITE awards are for applications using the PIC algorithm. Much of the national expertise for optimizing the basic PIC algorithm on current and future extreme computing platforms resides within the COMPASS project.

There is now a well established hierarchy of methods for modeling plasma based acceleration (many developed within SciDAC). Here we quickly review the hierarchy of methods. More details can be found in several references. The most "complete" method is the fully explicit, full Maxwell solver, fully relativistic, PIC method [Birdsall91]. In this method, a large number of macro-particles are initialized with positions within a grid and momenta. Each timestep, the currents (and in some cases the charge density) are "deposited" onto the appropriate corners or segments of the grid. Maxwell's equations are then advanced forward to calculate new electric and magnetic fields. This is most commonly done using the finite difference time domain (FDTD) method for which the fields need to be appropriately centered with respect to the cell (some are located at corners, some at the midplane of the cell boundaries, while others are defined in the middle of the cell) to achieve second order accuracy. This is commonly done using the Yee mesh. The fields can then be used to calculate the force on each particle so that their momentum and position can be advanced forward to the next time step after which the current can be determined and the whole sequence can be repeated. This "loop" can then be repeated the desired number of timesteps. The cell size is chosen to resolve the smallest scale length of physical relevance and the time step is then determined from the Courant condition. The shortest length of relevance is usually the laser wavelength for a laser driver and the wake's wavelength for a particle beam driver. In some cases the charged particles can radiate x-rays as they are accelerated. The wavelength of the x-rays is far too short to be resolved, and can be handled by secondary algorithms. While this algorithm is straightforward to explain, production codes are now very complex due to parallelization, single core optimization, various choices for how to deposit the current, higher order particle shapes, complex boundary conditions, diagnostics, and physics packages required to realistically model phenomena of interest.

The next method within the hierarchy takes advantage of the fact that a short pulse laser does not backscatter. In this limit, the laser pulse can be separated out into a central frequency (corresponding to a forward moving wave) and an "envelope" which is a complex number describing the pulse shape and length. Only the evolution of the envelope (both amplitude and phase) for the laser is calculated

and it is explicitly assumed that the time scale that the laser evolves over is much longer than a laser period (in fact longer even than a plasma wake period). Therefore, the time steps and grids used to advance the laser envelope can now be larger than the laser period and wavelength. The particles still oscillate at the laser frequency and wavenumber. However, one can further separate out the fields and forces into plasma wakefields and laser fields and then average over the laser frequency and wavenumber (assuming there is no backscatter). The result is that the force on the particles has the usual Lorentz force from the plasma wakefields and a ponderomotive or light pressure force from the laser. This is idea is called the ponderomotive guiding center (PGC) approximation [Mora97, Gordon00]. . The potential savings is very large. The cell size can be $\sim\omega_o/\omega_p$ larger since now the wake and not the laser wavelength needs to be resolved. The time step is also larger by the same factor (the Courant condition sets the time step to be the cell size divided by the speed of light). Since one wants to run the simulation for the same physical distance then the number of required time steps is therefore reduced by $\sim\omega_o/\omega_p$. Assuming the same number of particles per cell are used and the speed of the algorithm per particle step remains unchanged then this technique can lead to a speed up of $\sim(\omega_o/\omega_p)^2$. As we will show later such speed ups are being obtained. One potential issue is that as the laser propagates deeply into the plasma it pump depletes. As this occurs the central frequency decreases. Therefore over time the envelope approximation breaks down and methods to get around this issue are currently under development [Cowan10]. The actual speed up can be lower since one might need more particles per cell in the PGC because the cells are large, and because as the longitudinal cell size is reduced it becomes similar to the transverse cell size such that the actual time step is reduced by factors of $2^{1/2}$ and $3^{1/2}$ because of the difference in the Courant condition in 2d and 3d respectively. In addition, the PGC averaging procedure breaks down for very intense pump strengths.

Another method that takes advantage of the fact that there is no backscatter was recently proposed [Vay07]. In this method, one does the simulation is a Lorentz boosted frame (Figure 2). In a Lorentz frame moving at a speed near the speed of light with the laser, the laser will appear Lorentz expanded by a factor $(1+v_f/c)\gamma_f$ where $\gamma_f=(1-v_f^2/c^2)^{-1/2}$ and v_f is the velocity of the frame and c is the speed of light. The plasma (now moving towards you) is now Lorentz contracted (by a factor γ_f). In this frame the shortest wavelength is still the laser wavelength (so long as there is not light reflected as this light will have a very short wavelength in this frame) which has also been Lorentz expanded, and the number of wavelengths within the laser is a Lorentz invariant, i.e., it is the same in every reference frame. So although the laser is longer the number of cells required to resolve it does not change and the number of cells used for the simulation does not change. In physical units the time step is now larger [by $\sim(1+v_f/c)\gamma_f$]. The required number of time steps is therefore smaller by $(1+v_f/c)^2\gamma_f^2$ [a factor (1+v/c) arises because the pulse and plasma are moving towards each other]. The optimal frame to transform into is one moving near the group velocity of light within the plasma, $v_g=(1-\omega_p^2/\omega_o^2)^{1/2}$. In this frame $\gamma_f=\gamma_g=\omega_o/\omega_p$ such that the speed up once again scales as $\sim(\omega_o/\omega_p)^2$. And just like in the PGC method, the speed up is not quite as high as the above estimate since in the group velocity frame, the cells become nearly squares or cubes in 2d and 3d so that the time step needs to be slightly smaller. During the past year, substantial progress on this technique has been made [Martins10a, Martins10b, Huang09, Vay10]. This progress has involved working around a numerical instability that is still not completely understood. A sweet spot has been found that requires substantial smoothing which, based on convergence tests, appears to be fine for large γ_f . This technique is attractive because in principal it should work for full pump depletion distances and for arbitrary pump strengths. However, one must use smaller γ_f when investigating self-trapping, that is when the wake gets so large that background plasma electrons become self-injected into the wake and are accelerated. This is because the plasma is represented by γ_f fewer particles which can lead to a statistical issue. The PGC method has also been used for self-trapping but with its own limitations mentioned above. In addition, thinking about the physics in different inertial frames can help one better understand the physics.

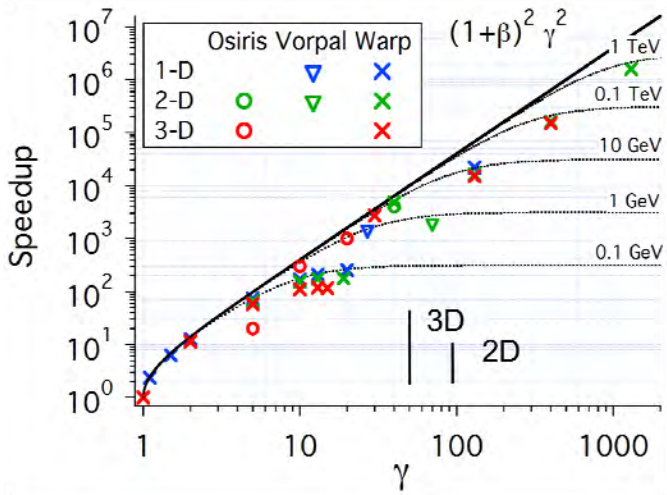


Figure 2. Plot of speed ups obtained do date using the boosted frame technique. Note that the OSIRIS simulations were for self-trapping and beam loading simulations using high intensity lasers. The Warp simulations were for test particles in a quasi-linear regime.

The last method in the hierarchy is the quasi-static approximation (which is used together with the PGC when using a laser driver). The quasi-static approximation [Mora97, Huang06] exploits the fact that during the time it takes the laser or particle beam driver to pass by a plasma electron the laser and particle beam appear static, i.e., neither their shape nor, their energy distribution (for a laser, this the frequency content) change. Under this approximation, one can calculate the wake for a given driver shape and then use the potentials and fields within this wake to get source terms to advance the laser or particle beam forward with time steps that resolve the time scale that the driver evolves over. These time scales are much larger than a plasma period. A laser driver typically evolves over a Rayleigh length, $k_0\, w_0^2/2$ while for a particle beam evolves over the betatron period $k_p/(2\gamma)^{1/2}$ (the period that particles oscillate transversely in the wakefield). For a laser driver the speed up on paper could therefore be the product of a factor of $\sim(\omega_o/\omega_p)^2$ from the use of PGC and an additional factor of $\sim(\omega_o/\omega_p)k_p^2w_0^2/2$, i.e., $\sim(\omega_o/\omega_p)^3k_p^2w_0^2/2$ from the quasi-static approximation. However, accuracy and numerical stability issues keep the time step for the laser solver to $\sim\sim(\omega_o/\omega_p)k_p\Delta$ where Δ is the transverse cell size. For a particle beam driver the speed up scales as $(2\gamma)^{1/2}$. Furthermore, the algorithm requires a predictor corrector loop so the field solve and particle push has additional overhead. This technique has already been used to verify some scaling laws out to 100 GeV [Tzoufras08b, Geddes08. Huang09]. An advantage of the quasi-static model is that it is not susceptible to numerical Cerenkov radiation so smaller transverse cell sizes can be used for narrow trailing bunches. In addition, mesh refinement should be easier to implement in principle. Calculating radiation reaction is also more straightforward. A disadvantage is that no prescription for handling self-trapping is presently available.

As the above indicates there is no one code and algorithm that fits all for modeling plasma-based acceleration. Currently, there are a suite of production codes that have been supported under the advanced accelerator part of SciDAC, OSIRIS [Fonseca02], VORPAL [Nieter04], and QuickPIC [Huang06]. In addition, another code Warp has been supported for electron cloud work. It has recently been used to develop the boosted frame technique and is starting to be used by some in SciDAC for LWFA simulations. OSIRIS is a full PIC code that also can run using the boosted frame technique primarily to study self-injection. VORPAL is a full PIC code that also can operate with the PGC, with cut cells for complicated boundaries, and is currently being updated to run using the boosted frame. Warp has been used to run laser wakefield simulations using the boosted frame technique (for very high γ in the external injection and weak beam loading regime). QuickPIC is a 3d quasi-static code which is used for both laser and particle beam drivers.

Software verification is an important part of the SciDAC mission. Previously we have reported comparison between all of these codes on a simple set of laser wakefield simulations [Paul08]. In these simulations the plasma wakefields were compared early in time before the laser had evolved.

The laser intensity was varied. The codes OSIRIS, VORPAL (with and without the PGC), Warp, and QuickPIC, and VORPAL all give very similar results. Comparison for longer run times will be undertaken in the following year.

3. Parallel Scalability

For extreme computing, it is not only necessary to have a code or algorithm, it is also important that the code scales well on parallel platforms. There are two types of parallel scaling studies. In strong scaling the problem size remains fixed as one increases the number or cores. In weak scaling the problem size increase in proportion the number of cores so that the size per core remains fixed. For many situations the weak scaling is most relevant since one typically uses extreme computing for large problems. This is generally true for plasma based acceleration. However, the strong scaling study is useful to understand the limitations of ones algorithm. The codes used under this project scale almost perfectly for weak scaling. Each of the codes also scales well for strong scaling so long as the number of particles per core is greater than ~100,000 and the domain size on a node is greater than ~4000 cells. The scaling will depend on how optimized the code is on a single core.

Examples of some recent results include strong scaling studies of OSIRIS on the Jugene Bluegene machine in Germany and the Jaguar computer at Oak Ridge. On Jugene OSIRIS had 81% efficiency out to 294912 cores (entire machine) with 1892 cells and 56000 particles per core at the end. On Jaguar OSIRIS had 60% parallel efficiency for 128000 cores with 4000 cells and 64000 particles per core at the end.

4. Results

In this section we describe some recent results and discoveries obtained from using OSIRIS, VORPAL, Warp, and QuickPIC. This includes helping to provide the scientific basis for two recently approved facilities dedicated to researching issues related to extending plasma-based acceleration towards a collider at the energy frontier.

4.1. Science Discoveries

An important issue for plasma based accelerators is how to simultaneously inject large amounts of charge into a plasma accelerator stage while maintaining high beam quality (emittance and energy spread). One issue with self-injection is that electrons which are injected also have large transverse momentum [Tsung04, Geddes05, Lu2007]. Several of the discoveries are related to understanding how to control injection. Such an understanding may ultimately lead to the injection and acceleration of high quality beams with significant charge. In addition, in the near future there may be 250 Joule short pulse lasers available. Another discovery is related to understanding how choices in the pulse length and spot size could affect the possible output energy and charge from a single stage.

4.1.1. Understanding the acceleration of self-trapped electrons. Experiments and simulations demonstrated in 2004 the formation of the first narrow energy spread and narrow divergence beams from laser plasma accelerators [Geddes04, Mangles04, Faure 04, Tsung04, Pukhov02]. Initial simulations described the essential physics of beam formation from self-trapped electrons, and the physics and optimization of these beams continues to be an important topic. Additional work has led to improved numerical techniques which control unphysical momentum spread [Cormier-Michel08], and quantitative agreement with the experiments [Tsung06, Geddes09] which together with advanced visualization [Ruebel10] allow detailed analysis of the physics of these stages (Figure 3). Tracking particles through the simulations shows that they were injected from the sides through the strong fields of the nonlinear wake, increasing their transverse momentum and hence the beam divergence undesirably [Tsung04, Geddes05, Lu07, Ruebel10]. Simulations also modeled GeV self trapping experiments in cm-scale capillary plasmas [Leemans06; Geddes09], showing that this process

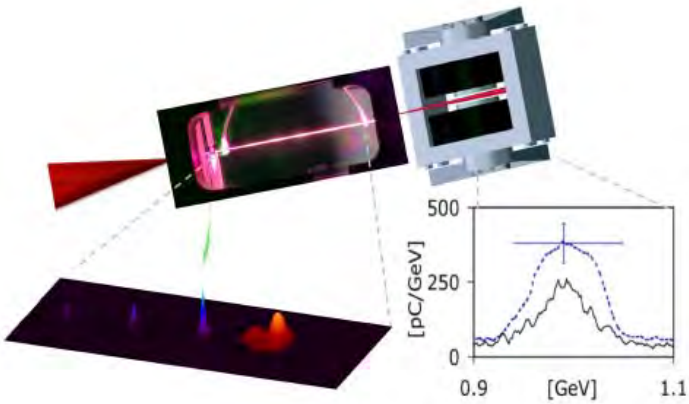


Figure 3. VORPAL simulations model formation of GeV electron beams from LBNL capillary experiments (top) showing the laser-driven wake and particles (left) and reproducing the experimental beam (right) [figure from Geddes09].

scales in a well understood way with plasma density. Such results are motivating next-generation experiments and simulations to control injection of electrons to further stabilize and improve beam quality.

4.1.2. Ionization trapping. One possibility for reducing the transverse momentum of self-injected electrons is to use ionization induced injection [Oz07, Pak10]. For a particle to be injected the change the wake must give it a change in potential that approaches ~mc^2/e. This is challenging if the particle originates outside the wake where the initial potential is 0. However, if the particle can be born inside the wake at a phase (location) where the potential is near a maximum (positive value) then it is easy for the particle to get trapped. It is easier to trap electrons near the axis and one can control where injection occurs by varying the gas mixture along the propagation direction. This allows one to control the emittance and energy spread. This was originally investigated for particle beam drivers [Oz07]. Recently, this idea was extended to laser drivers where the laser field also leaves an imprint on the particles transverse momentum [Pak10]. In the experiment of Pak et al. a mixture of He and N was used. Electrons from the K shell of nitrogen were tunnel ionized near the peak of the laser pulse and were injected into and trapped by the wake created by electrons from majority helium atoms and the L shell of nitrogen. The spectrum of the accelerated electrons, the threshold intensity at which trapping occurs, the forward transmitted laser spectrum, and the beam divergence are all consistent with this injection process. In the experiment it is not possible to conclusively show which shell an electron came from. However, this is possible in simulations. Full-scale 3D OSIRIS simulations showed that the trapped electrons came from the K shell of Nitrogen and they verified the theoretical threshold. Results from OSIRIS for both a beam or laser driver are shown in Figure 4.

4.1.3. Controlling injection to improve beam quality and stability. Further improvement of bunch momentum spread and day-to-day accelerator stability are required for applications such as light sources and high energy physics. Simulations are supporting experiments to develop controlled injection of particles decoupled from the guiding and acceleration process (in contrast to past self trapped experiments) to improve beam quality using tailored plasmas and multiple laser pulses.

By focusing the laser in a decreasing plasma density gradient, beams with longitudinal and transverse momentum spreads an order of magnitude lower than previously observed were produced [Geddes08]. Simulations showed these bunches result from modulation of the plasma wave phase velocity and can function as injectors to improve stability and reduce energy spread of high energy LWFAs [Geddes08, Geddes09], and initial results on density tailoring in a capillary waveguide have

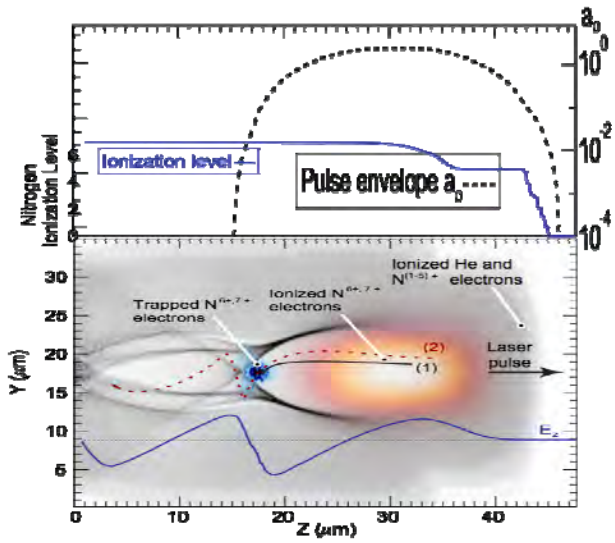


Figure 4a. In the top, a plot of the laser intensity envelope (log scale) showing the ionization level of the plasma mixture. In the bottom is a plot of the laser (orange) and wake (gray made up of mostly He electrons), the trajectory of a trapped (born near the axis) and untrapped K shell electrons as well as the axial wakefield (blue).

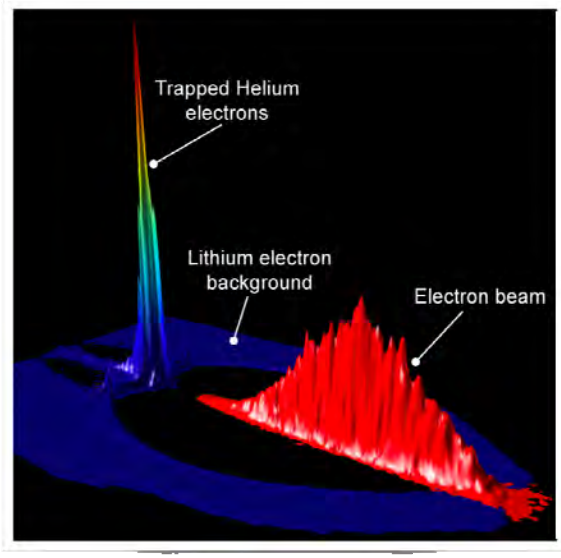


Figure 4b. A plot of the beam (red), the electrons make the wake (blue) and trapped electrons from Helium buffer gas. This OSIRIS simulation is for the E-167 experiment at SLAC.

demonstrated the increased stability predicted [Gonsalves07] and are continuing to tune energy spread and emittance, which is being supported by VORPAL envelope simulations which resolve trapping while allowing rapid 3D simulation required for these experiments.

Colliding laser pulses [Esarey97, Faure06] have been proposed and explored as an all-optical approach to producing stable, tunable, high-quality electron beams. The collision of two laser pulses creates a low velocity beat wave, which can give electrons an initial kick in velocity and position after putting them in phase to be accelerated by the plasma wake of the drive pulse. Self-consistent PIC simulations are important to characterize the dynamics of plasma electrons trapping into the plasma wake, in order to help maximize the trapped charge and minimize beam emittance and energy spread.

Simulations carried out with the parallel VORPAL framework are being used to guide and interpret experiments at the LOASIS Program of LBNL [E. Cormier-Michel10).]. Effects beam timing, angle, and intensity as well as plasma guiding were explored, demonstrating parameters that produce high quality beams. 2D simulations showed comparable results to 3D allowing parameter scans to be performed in 2D, reducing computational time and allowing for more thorough exploration of parameter space. Scaling laws were developed to understand scaling of laser and beam parameters with the plasma density and to allow rapid design of higher energy laser-plasma accelerators which use longer interaction lengths/lower densities.

The simulations are guiding parameters for colliding pulse experiments using the LBNL (0.6 J driver and <0.25 J collider) at 19° angle and show a beam of the order of 20 pC can be injected for a plasma density of ~4.10^{18} cm^{-3}. Recent experimental data have shown good agreement with simulation results including density and timing behavior (Figure 5), demonstrating the ability to quantitatively design such an experiment [Cormier-Michel08]. With optimized guiding of the laser pulse simulations indicate the beam could be accelerated to ~400 MeV with percent level energy spread, and effects of guiding and realistic laser mode have been characterized. Predictive verification against experiments provides confidence in the simulation scans being used to design future experiments to provide beam quality needed for applications.

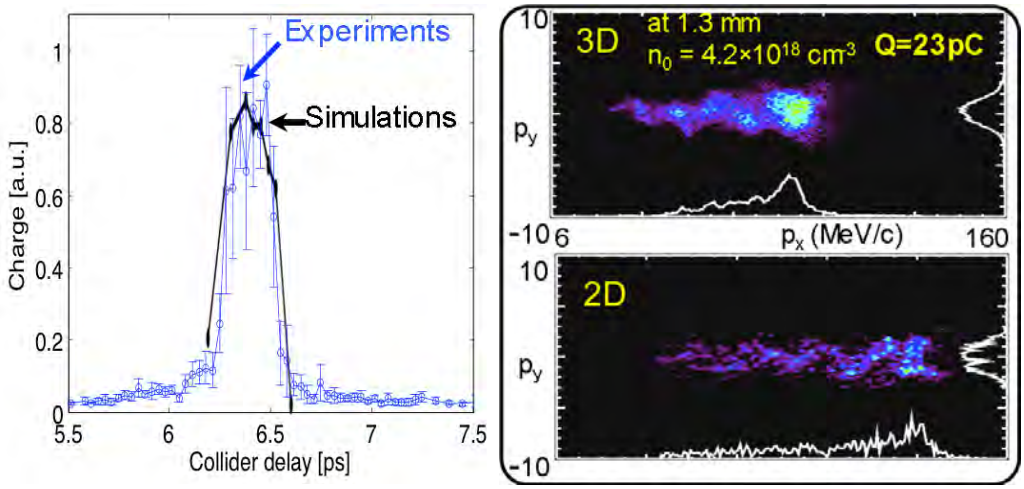


Figure 5. Simulations predicted parameters for colliding pulse experiments, including timing behavior (left), and demonstrate the formation of high quality beams (right, showing agreement of 2d and 3d results). Figure from Cormier-Michel et al, AAC 2008.

4.1.4. Next generation experiments. Since the results of Geddes04, Mangles04, Fuare04, and Tsung04, the past five years have seen an explosion of experimental results on generating mono-energetic electron beams from 100 MeV to 1 GeV [Leeman06, Clayton10]using a plasma wave accelerating structure driven by an intense laser (in fact there are too many to reference them all). The next decade will see tremendous increases in laser power and energy permitting parameter spaces to be accessed that could lead to bunches at the 10 GeV with .1 to 1nC of without external guiding. These new lasers will provide powers above a petawatt and energies at the 100s of Joule levels. They will also permit accessing the fully nonlinear regimes described in the recent theories.

We have been investigating through fully nonlinear particle-in-cell simulations the effects that choices in laser intensity and pulse length as well as plasma density can make for fixed laser energy in the output electron beam energy, charge, and quality [Martins10a]. In Figure 6 we show results from simulations where the laser energy was between 150 and 250 J. The simulation on the left was done in the lab frame (it simulated a 150Joule/30fs laser propagating in a 1.5 10^{19}cm^{-3} plasma, 2.3 GeV electrons were produced in a mm) while the simulation on the right (it simulated a 250J/220fs laser propagating in a 1.5 10^{19}cm^{-3} plasma, 40 GeV electrons were produced in 22cm) was only possible with the use of calculations in a Lorentz boosted frames We have also simulated several experiments including one at LLNL that recently observed 1 GeV quasi mono-energetic electron beams. The results are consistent with the predictions of Lu et al [Lu:07].

4.2. Supporting next generation facilities: BELLA and FACET

4.2.1. BELLA. Next-generation experiments at the BELLA laser facility under construction at LBNL will use controlled injection coupled with meter-scale plasmas to increase bunch quality and increase energy to 10 GeV. A future collider could use many modules in series, each increasing the beam energy to reach TeV energies [Leemans09]. These stages must preserve low energy spread and good emittance for electrons, and also for positrons motivating study of less strongly driven, quasilinear wakes, where dynamics are similar for electrons and positrons and where the shape of the wake is controlled by thelaser profile. With simulation of present cm-scale experiments pushing state of the art computing, new techniques are required. Scaling of the design with plasma density, envelope simulations and Lorentz boosted simulations have been combined to model these stages.

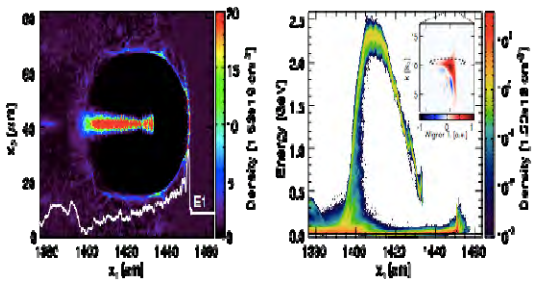


Figure 6a. Results for using a tightly focused 150 J laser at high density. On the left is the plasma density and the trapped particle density as well as a lineout of the accelerating field. On the right is the energy vs. position of the trapped particles. Note that the integrated energy spread is large while the slice energy spread is small.

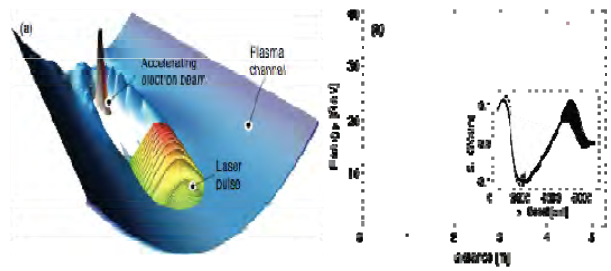


Figure 6b. Results for a wide focus and long pulse regime for a 250J laser propagating in a channel. On the left is laser pulse (yellow and red) propagating in a channel. The blue is the plasma density and the fin in the back is the trailing beam. On the right is the energy vs. time for the beam and the lineout of the acceleration field.

Theory predicts that wake structure will scale with plasma density but does not predict the shape of the wake or how the plasma will focus the laser. A series of VORPAL PIC simulations characterized wake structure and evolution and verified predicted scaling with density, allowing scalable design [Cormier-Michel08]. Hence a short simulation at high density predicts performance for high energy stages and also provides understanding of how stage performance will scale with laser and plasma performance. Variation of laser pulse amplitude, width and length together with plasma longitudinal taper (for beam phase control) and transverse profile (for guiding) established parameters that best deplete the laser energy into the accelerating field of the wake while maintaining quasilinear structure. Electron beam charge, length, and width were next adjusted to obtain efficient acceleration—that is, high transfer of laser energy deposited in the wake to the particles resulting in a design for 10 GeV stages with the Petawatt BELLA laser.

Some parameters such as electron bunch oscillation in the focusing field do not scale with density, requiring simulation of the full meter-long stage at 10^{17}/cc densities, and this in turn requires advanced models which reduce computational load. Over 4 orders of magnitude speedup has been obtained by modeling the laser as a complex scalar envelope modulating a plane wave, enabling unscaled, 3D simulations of full meter-scale stages (PGC code). Recently, the ability to track separately a local laser phase has allowed this envelope model to correctly resolve the laser field into significant depletion. This development is being continued to allow the envelope model to propagate into deep depletion [Cowan10].

Lorentz boosted frame simulations using WARP have been used to verify the scaling of energy gain from 100 MeV to up to 1 TeV [Vay10] on simulations of externally injected low charge beams, thanks to speedups ranging respectively from 100 to over 1 million times (compared to explicit PIC in the laboratory frame). At 10 GeV, the maximum attainable speedup of 10,000 times was achieved, speeding up the modeling of one BELLA stage with explicit PIC from years to hours (assuming a few thousands of CPUs on present day supercomputers) (Figure 7).

Combined simulation with scaled, envelope and boost techniques have been used to detail understanding of 10 GeV stages with few hundred pC charge and low energy spread for both electrons and positrons using a petawatt laser, and how such stages scale for other application such as GeV light sources stages. This checks the models against one another for accuracy, and derives physical

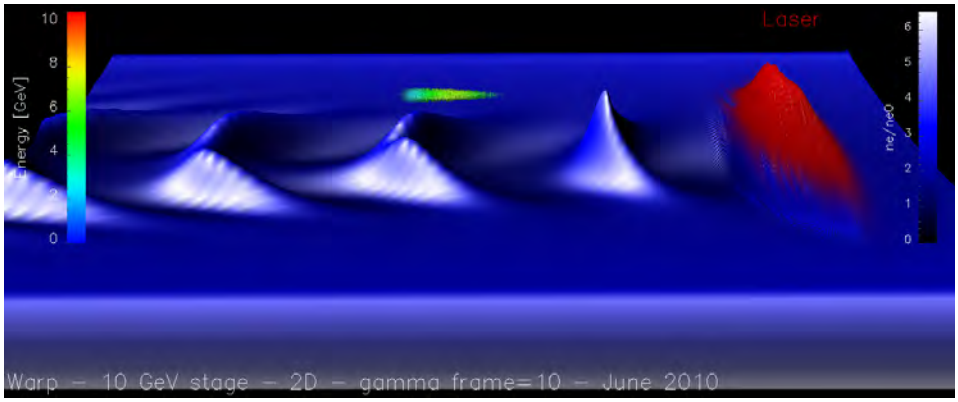


Figure 7. Image from a WARP simulation of a strongly depleted 10 GeV BELLA stage in a Lorentz boosted simulation frame showing the wake (blue) driven by the laser (red) and the accelerated particle bunch.

understanding from the differences each approximation reveals. Future simulations will use the improved computational efficiency demonstrated together with new computing resources to increase resolution to capture and tune the stages to preserve the very low emittances required for applications, and staging to the energy frontier.

4.2.2. FACET. Next generation experiments at the FACET facility which is under construction at SLAC will be aimed at showing that a trailing bunch of electrons can be accelerated with high transfer efficiency while maintaining its emittance and energy spread. The FACET facility will have electron and positron bunches at 23 GeV with 3nC of charge. In addition, the beams can have bunch lengths as short as 14μm which corresponds to peak currents of 22 kA. Designs of these experiments indicate that two bunches with ~1nC in the drive beam and ~.5nC in the trailing beam and a separation ~100 μm are possible. These conditions have been simulated with QuickPIC to find the optimum density. Other beam parameters have also been simulated. In Figure 8 we show that for the expected conditions, that the trailing beam can be accelerated from 23 GeV to 50 GeV with a few % energy spread in less than 1 meter. The QuickPIC simulations are an essential tool in the design of these experiments.

4.3. Issues for Colliders

The high fidelity SciDAC codes are critical for assessing critical issues related to a future collider. Issues related to emittance preservation for collider parameters, the use of asymmetric beams, staging, optimal beam and laser profiles, and the possibility of higher power lasers and beams cannot all be addressed at FACET or BELLA. They can be addressed using trusted codes and extreme computing. In this section we give two examples of how computing is addressing issues beyond BELLA and FACET.

The ultra-low emittance beams required for next-generation lepton colliders and the large focusing forces in typical plasma accelerators imply that the matched spot size of the electron beam is much smaller than the plasma wake width. For linear wakes this can limit bunch charge and hence the efficiency of a plasma accelerator, because a small bunch with high charge modifies the plasma fields causing nonlinear emittance growth. This problem can however be mitigated by using nonlinear wakes [Tzoufras:09] (which can lead to other issues) or by tailoring the focusing fields in the linear regime through the use higher order laser modes. Simulations with the VORPAL framework have shown that it is possible to co-propagate different laser modes in a plasma channel up to the electron beam dephasing length, keeping the focusing forces small during the whole acceleration process, and that

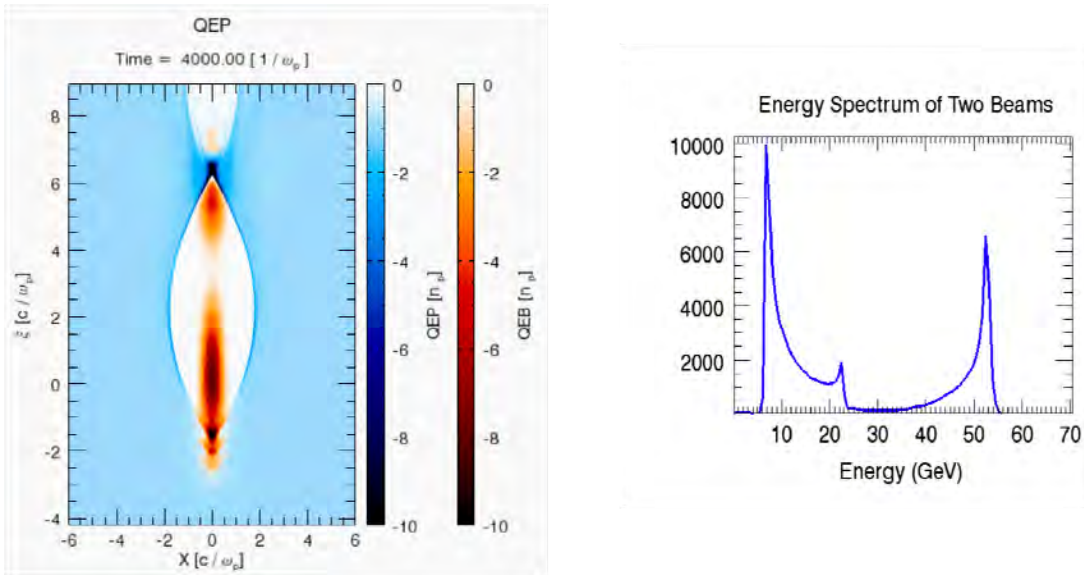


Figure 8. QuickPIC simulations of possible two bunch experiments at FACET. On the left is the beam and plasma density. The plasma density is chosen so that the trailing bunch loads the rear of the wake. On the right is the energy spectra of both bunches. The drive bunch is been decelerated from 23 GeV to 6 GeV while the trailing bunch has been accelerated to 50 GeV.

phasing between the modes can be used to control the focusing phase and field gradients. Simulations with test particles have shown that the matched beam spot size can be increased by almost a factor of 3 using this method, while keeping the low emittance of the beam (Figure 9) [E. Cormier-Michel10].

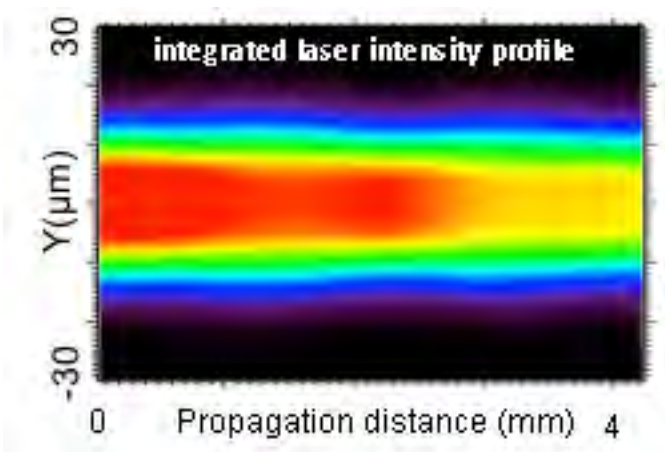


Figure 9. VORPAL simulation shows that multiple laser modes can co-propagate up to the depletion & dephasing distance required for an efficient accelerator while maintaining a flat transverse (Y) profile to tailor focusing forces [Figure from Cormier-Michel et al, sub PRSTAB].

A PWFA-LC design [Seryi09] uses a conventional 25 GeV electron drive beam accelerator, to produce trains of drive bunches distributed in counter-propagating directions to 20 PWFA cells for both the electron and the positron arms of the collider to reach energy of 500 GeV for each beam. Each cell provides 25 GeV of energy to the main beam in about a meter of plasma. The layout and parameters were chosen to optimize PWFA performance while also providing feasible parameters at the interaction point and a practical design for the main beam injector and the drive beam acceleration and distribution system. The wakes are to be excited in a nonlinear or weakly nonlinear blow-out regime where if the plasma ions remain stationary a uniform ion column results which provides a linear focusing force and a radially independent focusing force for both the driving and accelerated

electron beams. There is a small region in a weakly nonlinear wake that can both focus and accelerate positrons.

Properties of the drive and main beam bunches have been optimized by particle-in-cell simulations using the code QUICKPIC [Huang:06]. Simulation results are shown in Figure 10. The main beam bunch charge is 1.0×10^{10} particles with a Gaussian distribution. A plasma density of $10^{17}cm^{-3}$ and a Gaussian drive bunch charge of 2.9×10^{10} were chosen, the simulations showed 48% power transfer efficiency from the drive beam to the main beam with a gradient of roughly 25 GV/m. The drive beam bunch length is 30 μm while the main beam bunch length is 10 μm and the drive-main beam bunch separation is 115 μm. The separation between the two bunches must be approximately equal to the plasma wavelength. The energy spread was maintained at less than 2%. We have also developed theories for how to tailor the current profile of the drive and main beams to achieve higher transfer efficiencies and smaller energy spreads [Tzoufras:08a]. QuickPIC simulations show that lower energy spreads are possible (less than 1%) but that hosing could potentially limit the energy transfer efficiency for the longer bunches. Hosing is also an issue for LWFA.

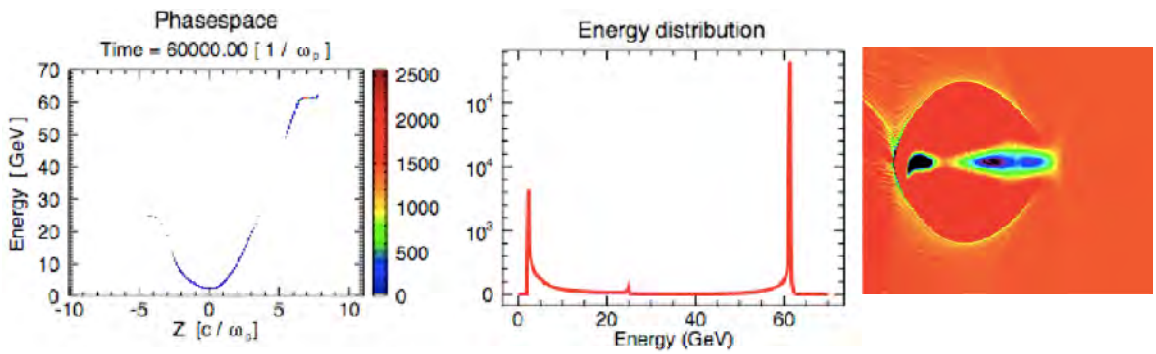


Figure10. QuickPIC results for a single stage in a PWFA-LC design. Using non-optimized Gaussian bunches gives 48% transfer efficiency while maintaining ~1% energy spread. On the left is the energy vs. z. In the middle is the energy spectrum. On the right is the beam and plasma densities. In each case both the drive and trailing beams are shown.

5. Summary and Some Thoughts on What is Next

Simulations have played a critical role in the development of plasma based accelerators. They are now routinely used to design and interpret state-of-the-art experiments. They were an essential component for making the science case for two recently approved experimental facilities, BELLA and FACET. They have helped make science discoveries and are being used to study what is possible for laser facilities being considered at the 250 Joule energy level. They are also currently being used to address issues for colliders based on plasma accelerator stages under consideration as next steps beyond BELLA and FACET.

Simulation tools continue to improve. Within the COMPASS project there is a hierarchy of codes and methods. These methods include full PIC, full PIC in a boosted frame, full PIC with ponderomotive guiding center (PGC), quasi-static PIC with PGC, and fluid models. Previously, the quasi-static technique had allowed the study of 1-100 GeV LWFA stages [Tzoufras08b, Huang09] and upto 1 TeV PWFA stages. In the past year, there has been tremendous progress in the use of each. For self-injected cases boosted frame simulations give speed ups ~300 and for external injection speedups $\sim(\omega_o/\omega_p)^2$ are possible making it possible to model TeV LWFA external injected stages. PGC simulations are also giving speed ups $\sim(\omega_o/\omega_p)^2$. The next year will see further comparison between all techniques.

The future is even more exciting, as we are about to enter the era of exascale computing. The new systems are likely to share some of the following characteristics: a heterogeneous hierarchy of powerful shared memory multi-core nodes, with some element of SIMD (vector) processors as part of their design, coupled together with traditional message passing networks. To prevent memory starvation, they will likely have lightweight threads which can hide memory latency. To meet this challenge for PIC codes, new data structures and algorithms will be required. These algorithms will need to be parameterized so they can work on any future architecture that has these elements.

New kernels have already been implemented for key elements of the PIC algorithm [Decyk09, Decyk10]. For example, a kernel for a 2D electrostatic code that contains 6 subroutines (including a spectral field solver) for a 2D electrostatic PIC code has been tested on a NVIDIA GTX 280 GPU. For an optimum case where the particles never leave a cell the speed up was 44, while for a realistic case (typical plasma temperature) the speed up was 28 (1 ns per particle advance) over an Intel i7 CPU. From this performance it can be estimated that a 3D electromagnetic code could achieve speed ups in excess of 100 (<3ns per full particle advance) for currently used algorithms. A 2D EM kernel for a rigorous charge conserving algorithm based on a different data structure and sorting routine has achieved speed ups from 27-81 depending on the initial plasma temperature [Kong10].

While current GPUs appear to offer the greatest acceleration, one should expect that not only will they evolve significantly but future many core platforms will also evolve rapidly. As a result algorithms will need to be retuned regularly. Therefore, as noted above new algorithms should be parameterized as much as possible. It should also be noted that on current systems there is usually one GPU per 4 cores so speed ups of only 10 are not competitive with SIMD units for which there are usually one per core.

If one assumes speed ups ~100 for next generation machines per core and similar numbers of cores and that additional speed ups of ~100 from existing algorithms are possible (for example the quasi-static could produce this) then the turnaround time of simulations for next generation experiments could be ~minutes or less. This could lead to real time steering of experiments as well as the ability of using simulations to design and optimize laser, beam, and density profiles including energy chirps.

Acknowledgments

Work supported by the U.S. Department of Energy (DOE) Office of Science, Office of High Energy Physics SciDAC program and under grants including DE-AC02-05CH11231, DE-FG03-95ER40926, DE-FG02-01ER41178, DE-FG02-03ER83857, DE-FC02-07ER41500; and by the National Science Foundation under the grants grants PHY-0904039 and PHY-0936266**.** This work used the resources of the National Energy Research Scientific Computing Center, the Hoffman 2 cluster at UCLA, and at the National Center for Computational Sciences.

References

[Abraham03] S. Abraham, Spencer Abraham announces department of energy 20-year science facility plan, text available at http://www.er.doe.gov/Sub/Newsroom/News_Releases/DOE-SC/2003/Energy%2020-Year%20Science20Facility%20Plan.htm, 2003.

[Bethel09] E.W. Bethel, et al. "Occam's Razor and Petascale Visual Data Analysis," Journal of Physics: Conference Series 180, 012084 (2009).

[Birdsall91] C. K. Birdsall, A. B. Langdon, V. Vehedi, and J. P. Verboncoeur, Plasma Physics via Computer Simulations, Adam Hilger, Bristol, 1991.

[Blumenfeld07a] I. Blumenfeld et al., Nature 445, 741 (2007).

[Clayton10] D. Froula et al., Self-guided laser wakef eld acceleration beyond 1 GeV using ionization-induced injection, to appear in Phys. Rev. Lett.

[Cormier-Michel08] E. Cormier-Michel, et al., "Unphysical kinetic effects in laser wakefield accelerators modeled with particle-in-cell codes," Phys. Rev. E. 78, 016404 (2008).

[Cormier-Michel10]E. Cormier-Michel, et al. "Control of focusing fields in laser-plasma accelerators using higher-order modes," PRSTAB submitted.

[Cowan08] B.M. Cowan, et al, "Characteristics of an envelope model for laser plasma accelerator simulation", J. Comput. Phys., submitted Computer Physics Communications.
[Decyk09] V. Decyk, T. Singh, and S. Friedman, Graphical Processing Unit-Based Simulations, in Proc. Intl. Computational Accelerator Physics Conf. (ICAP2009), 2009, [Decyk10] Viktor K. Decyk and Tajendra V. Singh, submitted to
[Esarey96] E. Esarey et al., IEEE Trans. Plasma Sci. 24, 252 (1996).
[Esarey97] E. Esarey et al., Phys. Rev. Lett. 79, 2682 (1997).
[Faure04] J. Faure et al., Nature 431, 541 (2004).
[Faure06] J. Faure et al., Nature 444, 737 (2006).
[Fonseca02] R.A. Fonscea et al., Lect. Notes Comp. Science 2329, III-342, Springer-Verlag, (2002).
[Geddes04] C. G. R. Geddes et al., Nature 431, 538 (2004).
[Geddes05] C.G.R. Geddes et al. "Production of high-quality electron bunches by dephasing and beam loading in channeled and unchanneled laser plasma accelerators," Phys. Plasmas 12, 056709 (2005).
[Geddes07] C. G. R. Geddes et al., J. Phys. Conf. Ser. 78, 12021 (2007).
[Geddes08a] C.G.R. Geddes et al, Phys. Rev. Lett. 100, 215004 (2008).
[Geddes09] Geddes et al. "Laser Plasma Particle Accelerators: Large Fields for Smaller Facility Sources," SciDAC Review 13 (Summer, 2009); http://www.scidacreview.org/0903/index.html.
[Gonsalves07] A.J. Gonsalves et al., in Proc. Particle Accel. Conf., IEEE, Piscataway, NJ (2007).
[Gordon00] Daniel F. Gordon et al., IEEE Trans. Plasma Sci, Vol. 28, 1135 (2000)
[Huang06] C.K. Huang et al. Journal of Computational Physics 217 (2006) 658–679
[Huang09] C. Huang, et al., "Recent results and future challenges for large scale Particle-In-Cell simulations of plasma-based accelerator concepts", J. of Physics: Conference Series 180, 012005 (2009).
[Joshi06] C. Joshi, "Plasma accelerators," *Scientific American*, Vol. 294, 41-47 (Feb. 2006).
[Kong10] X. Kong, et al., "Particle-in-cell Simulations with Charge- Conserving Current Deposition on Graphic Processing Units," submitted to J. Computational Phys., 2010.
[Mangles04a] S. P. D. Mangles et al., Nature 431, 535 (2004).
[Leemans05a] W. P. Leemanset al., 2005 IEEE Trans. Plasma Sci. 33, 8 (2005).
[Leemans06a] W. P. Leemans et al., Nature Physics 2, 696 (2006).
[Leemans09] W.P. Leemans and E.H. Esarey, "Laser-driven plasma-wave electron accelerators," Physics Today 2009.
[Lu07a] W. Lu et al in Phys. Rev. ST Accel. Beams 10, 061301 (2007)
[Martins10a] Samuel F. Martins et al., "Exploring laser-wakefield-accelerator regimes for near-term lasers using particle-in-cell simulation in Lorentz-boosted frames," Nature Physics, **6**, 311–316 (2010)
[Martins10b] Samuel F. Martins et al., Computer Physics Communications, 181, 869-875 (2010).
[Michel08a] E. Cormier-Michel et al., accepted to PRE (2008).
[Mora97] P. Mora, T.M. Antonsen Jr., Kinetic modeling of intense, short laser pulses propagating in tenuous plasmas, Phys. Plasmas 4 (1997) 217.
[Nieter04] C. Nieter and J. Cary, J. Comp. Phys. 196, 448 (2004).
[Oz07] E. Oz, et al., "Ionization-induced electron trapping in ultra-relativistic plasma wakes," *Physical Review Letters*. 98, 084801(2007).
[Paul08] K. Paul, et al., "Benchmarking the codes VORPAL, OSIRIS and QuickPIC with Laser Wakefield Acceleration Simulations," Proc. Advanced. Accel. Concepts Workshop (2008).
[Pak10] Pak A., et al., "Injection and trapping of tunnel ionized electrons into laser-produced wakes, *Physical Review Letters*, Vol. 104, 025003 (January 2010).
[Pukhov02] A. Pukhov and J. Meyer-ter-Vehn , 2002 Appl. Phys. B 74, 355 .
[Rubel10] O. Rübel, et al., "Coupling Visualization and Data Analysis for Knowledge Discovery from Multi-dimensional Scientific Data," Procedia Computer Science 1, 1751 (2010).
[Tajima79] Tajima T. and Dawson J. M. , Phys. Rev. Lett. 43, 267 (1979)

[Tsung04] F. S. Tsung et al., Phys. Rev. Lett. 93, 185002 (2004) .
[Tsung06] F. S. Tsung et al., Physics of Plamas **13**, 056708 (2006)
[Tzoufras08a] M. Tzoufras et al., Phys. Rev. Lett. vol. (2008).
[Tzoufras08b] M. Tzoufras, PhD Dissertation UCLA 2008.
[Vay07] Vay J.-L., Phys. Rev. Lett. 98, 130405 (2007).
[Vay10] J.-L. Vay, et al., "Modeling laser wakefield accelerators in a Lorentz boosted frame," submitted to J. Comput. Phys.

Massively parallel simulation of anisotropic Debye shielding in the modulator of a coherent electron cooling system and subsequent amplification in a free electron laser

B.T. Schwartz,[1] D.L. Bruhwiler,[1] V.N. Litvinenko,[2] S. Reiche,[3] G.I. Bell,[1] A. Sobol,[1] G. Wang,[2] and Y. Hao[2]

[1] Tech-X Corp., Boulder, CO 80303, USA
[2] Brookhaven National Laboratory, Upton Long Island, New York, USA
[3] Paul Scherrer Institut, Villigen PSI, 5232, Switzerland

Abstract. Novel electron-ion collider concepts are a high priority for the long-term plans of the international nuclear physics community. Orders of magnitude higher luminosity will be required for the relativistic ion beams in such accelerators. Electron cooling is a promising approach to achieve the necessary luminosity. The coherent electron cooling (CeC) concept proposes to combine the best features of electron cooling and stochastic cooling, via free-electron laser (FEL) technology, to cool high-energy hadron beams on orders-of-magnitude shorter time scales. We present δf particle-in-cell simulations of a CeC modulator, using the parallel VORPAL framework. We also present GENESIS simulations relevant to subsequent amplification of the electron wake in a high-gain FEL.

1. Introduction

Increasing the luminosity of hadron accelerators is important for advancing the fields of particle and nuclear physics. One method of achieving this goal is coherent electron cooling (CeC) [4], which involves Debye shielding of moving ions in a plasma with an anisotropic velocity distribution. An analytic solution has been developed for analyzing this case of Debye shielding for a Lorentzian velocity distribution [8]. Here we use this analytic solution to validate particle-in-cell (PIC) simulations of ion shielding in plasmas with isotropic and anisotropic Lorentzian velocity distributions. Having validated the simulation, we apply a more realistic Maxwellian velocity distribution to cases with two closely-spaced ions, showing that the electron response is linear and, hence, additive.

1.1. Coherent Electron Cooling

The primary cooling mechanism in a standard electron cooler is dynamical friction on the ions (see e.g., Ref. [1] and references therein). The modulator section of a coherent cooler would be similar to a standard cooler, but in this case dynamical friction becomes irrelevant and the key physics is the coherent density and velocity perturbations imprinted on the electron distribution via anisotropic Debye shielding of each ion [4].

Coherent electron cooling consists of two components in addition to the modulator: a free-electron laser (FEL) amplifier, and a *kicker* [4]. In the modulator, electron and ion beams copropagate at the same velocity such that the ions imprint a modulation on the electron density. Both beams have highly anisotropic velocity distributions such that $\sigma v_{x,y} \gg \sigma v_z$.

We first consider the motion of a single ion in the beam reference frame. Electrons gather preferentially behind the ion, forming a cloud that shields the ion charge, with density dropping to the background level over several Debye lengths along each dimension. Since the transverse rms velocities exceed the longitudinal, the cloud resembles a pancake for a stationary ion. In the laboratory frame, the longitudinal dimension contracts by the beam's Lorentz factor γ_0, so the pancakes can be very thin.

After the modulator, the electron beam propagates through a high-gain FEL, which amplifies the pancake-shaped electron clouds that had formed in the wake of each ion, resulting in a longitudinal charge density variation with a period equal to the FEL wavelength. Each ion is phase shifted with respect to the electron beam such that copropagation in the kicker leads to an interaction with its own amplified electron wake, decreasing the relative velocity [4].

1.2. Debye Shielding

In isotropic Debye shielding of stationary ions, the steady-state electric potential $\phi \propto \frac{1}{r}$ around the ion decays exponentially as $\exp\left(\frac{-r}{\lambda D}\right)$, where $\lambda_D = \left(\frac{\epsilon_0 \kappa T}{e^2 ne}\right)^{1/2}$ is the Debye length [6].

However, this expression for shielding does not apply to the CeC case for two reasons: (1) the electron-ion interaction time is less than the plasma period, and (2) the electron beam temperature is highly anisotropic. Time-dependent analytical solutions for plasmas with Lorentzian velocity distributions are derived in Ref. [8]. In Sect. 2 we use this solution to validate particle-in-cell simulations of ion shielding in isotropic and anisotropic plasmas with Lorentzian velocity distributions. Having validated the simulation, we then consider a more realistic Maxwellian velocity distribution.

1.3. VORPAL δf PIC Simulations

The ion shielding simulations employ VORPAL's δf PIC algorithm, which uses particle weights to measure the perturbations away from an equilibrium state [9]. The δf PIC method provides significant computational advantages over conventional PIC computations. To obtain even moderate agreement with theory, conventional PIC simulations required volumes of 10^4 Debye cubes, with at least 1,000 particles per cell. These correspond to simulations with approximately 10^7 mesh cells and 10^{10} macroparticles, which took several hours to run on a supercomputer with 2,048 cores. By comparison, the δf PIC simulations shown in Sect. 2 typically required only 200 particles per cell, yielding better agreement and less particle noise, even with lower resolution. Simulations using 2048 cores on the Franklin supercomputer at NERSC typically require 90 minutes to simulate shielding for durations up to half a plasma period.

2. Software Validation

For the case of Debye shielding by electrons with a kappa velocity distribution with $\kappa = 2$ (i.e., a Lorentzian) [2], analytic results have been derived [8]. In this section we validate VORPAL δf PIC simulations for this case by comparing the results to these analytical predictions.

Equilibrium Debye shielding around a single Au^{+79} ion produces an excess of ~79 electrons among ~10^7 physical electrons in the relevant domain. Since the simulation domain is finite, the simulation is set up to allow a thermal flux of electrons to leave and enter the boundaries. Electrons near the boundaries that had just recently entered the simulation domain have errors in their positions and velocities, since they have only begun to experience the fields from the ion and other electrons. Hence, only the central portion of the domain is shown in the Figures 1–4. To obtain good agreement the post-processing did not evaluate the perturbed electron charge within a few Debye lengths (typically 5) of the simulation domain edge.

Figure 1 compares the local integrated shielding charge verses time for the linear analytical prediction and the VORPAL δf PIC simulation. In this case the electron velocity distribution is anisotropic such that $v_x = v_y = 3v_z$ ($R = 3$), where z is the longitudinal direction. The total perturbed

electron charge that accumulates in a finite volume around the ion is expected to peak at half a plasma period, after which the oscillation amplitude dampens, as shown in Figure 2. The plasma with a Maxwellian velocity distribution exhibits oscillatory behavior similar to the Lorentzian case, but with different amplitude and phase.

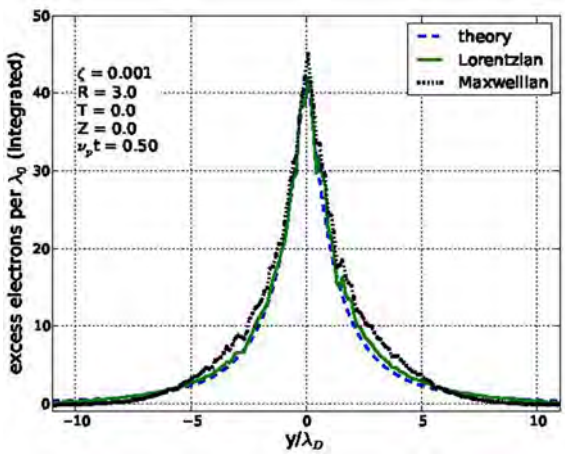


Figure 1. Longitudinal charge density perturbation in the vicinity of the Au^{+79} ion, for the case of a stationary ion in an anisotropic plasma with both Lorentzian and Maxwellian e^- velocity distributions.

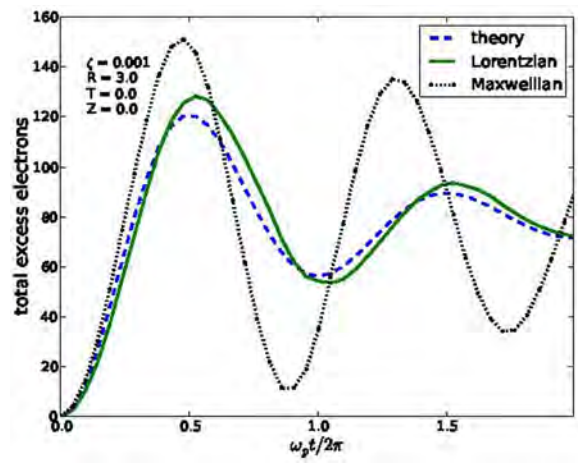


Figure 2. Time evolution of the integrated e^- charge enhancement in the vicinity of the Au^{+79} ion, for the case of a stationary ion in an anisotropic e^- distribution. The time scale is in units of plasma period.

Good agreement between the VORPAL simulations and theory is maintained for moving ions. The simulations in the following plots show Debye shielding for anisotropic plasmas with the ratio of transverse to longitudinal velocity spread of R = 3. Figure 3 and Figure 4 show the shielding charge for a gold ion moving in both the longitudinal (z) and transverse (x) directions. In each case the ion is moving with a transverse velocity of T = 5.4 times the plasma's rms transverse velocity and Z = 0.6 times plasma's rms longitudinal velocity. We note that the Maxwellian and Lorentzian responses can differ significantly, as seen in Figure 4.

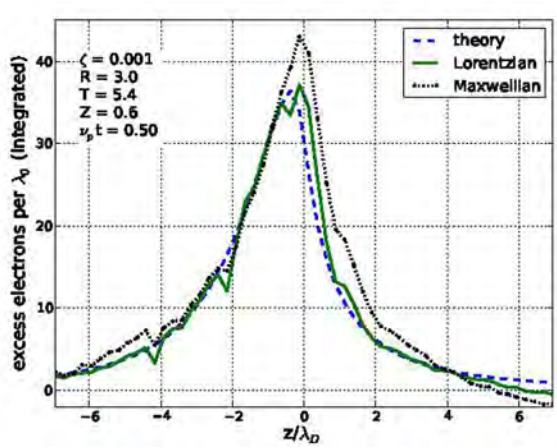


Figure 3. Longitudinal charge density perturbation of a plasma in the vicinity of a moving Au^{+79} ion.

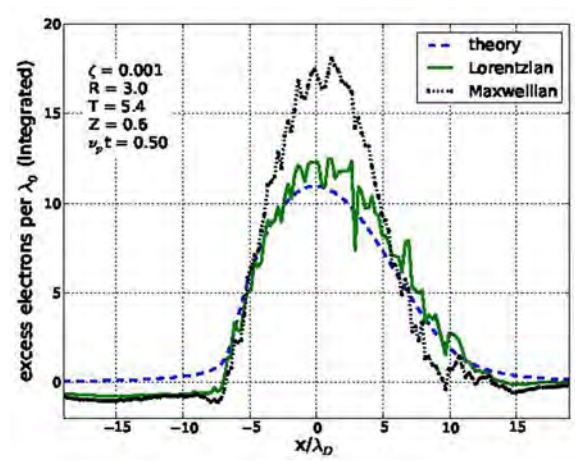


Figure 4. Transverse charge density perturbation of a plasma in the vicinity of a moving Au^{+79} ion.

The following quantities were varied to obtain good agreement of the δf PIC simulations and the theoretical prediction: domain width, number of cells per Debye length, number of particles per cell, and electron velocity range. The domain width ranged from $24\lambda_D$ to $48\lambda_D$, and was sampled with four to eight cells per Debye length. The particles per cell ranged from 200 in most cases, to 800 for the faster moving ions. Modeling such cases also required increasing the electron velocity range from $\pm 5v_{\mathrm{rms}}$ to $\pm 7v_{\mathrm{rms}}$.

3. Debye Shielding of Two Ions

It is important for CeC that the plasma's response to an ion beam be linear, meaning that the total response is approximately the superposition of responses to each ion. Otherwise individual ions will not be properly accelerated or decelerated (cooled) in the kicker. Our initial inquiries into this matter involve electron shielding simulations of two ions separated by different distances. Figure 5 shows a case where plasma has an anisotropic Maxwellian velocity distribution, $R = 3$. The number of shielding electrons is essentially independent of separation distance; the variation was less than 2%. Hence, the response is linear.

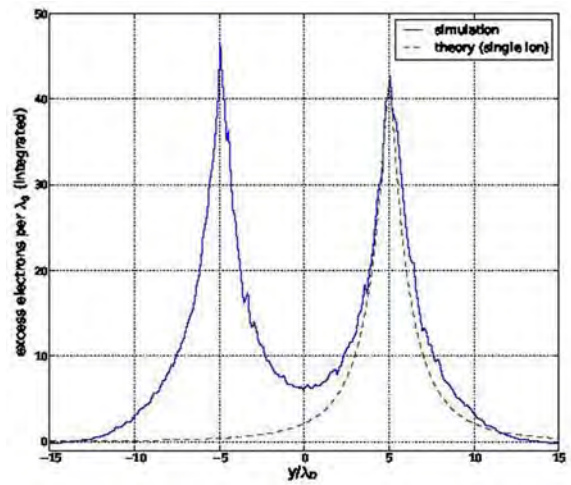


Figure 5. Transverse charge density perturbation of a plasma in the vicinity of two stationary Au^{+79} ions separated by $10\lambda_D$. Dotted line: theoretical prediction for a Lorentzian velocity distribution.

4. Free-Election Laser Amplification of the Electron Wake

After the modulator, the electrons pass through an FEL, which amplifies the electron distribution and modulates it with a period equal to the FEL wavelength. Correct simulation of the coupling from modulator to FEL amplifier involves extracting bunching parameters and associated phases from the electron density histograms shown above in Sect. 2, and using them to properly modify the input electron distribution used by GENESIS [7], an FEL simulation code. We will present the details of such calculations in future work.

As an intermediate step toward converting λf VORPAL outputs to inputs for GENESIS, we used GENESIS to simulate a $\lambda_0 = 700$ nm FEL with an electron beam that had a non-zero bunching parameter b only in the leading 1% if its length. This bunching was sufficient to cause lasing via self-amplified spontaneous emission for bunching parameters as small as $b = 10^{-10}$. Figure 6 shows that the FEL imparts a charge density oscillation on the electron beam.

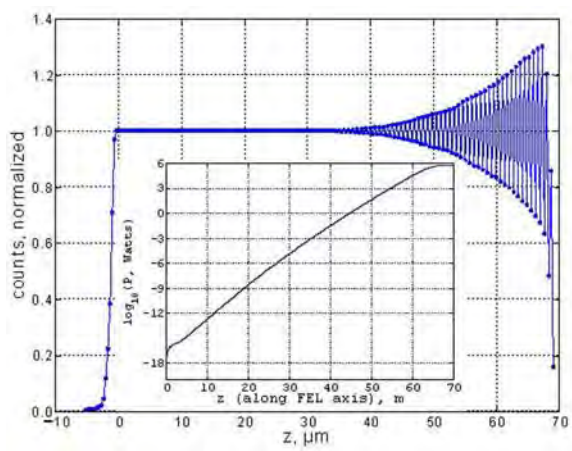


Figure 6. Histogram of electron beam particle counts after propagating 60 meters in an $\lambda_0 = 700$ nm FEL, bin width $= \frac{\lambda_0}{2}$. Inset: Optical power along FEL axis.

Using an algorithm based on Refs. [3] and [5], we found the bunching magnitude extracted from this signal at different positions in the FEL to agree fairly well with those computed within GENESIS. To verify correct bunching, we

employed an algorithm identical to the standard definition of bunching and method used in GENESIS: the average the electron ponderomotive phase in each λ_0-wide slice of the electron distribution. Both the bunching magnitude and phases computed using this method agreed very well with the GENESIS calculations.

5. Scaling of λf PIC Simulations on Many Processors

Figure 7 shows the strong scaling of the simulation time for the λf PIC ion shielding simulations shown in Figures 3 and 4. Removing the ion (Z = 0) improved scaling slightly, as the reduced electron motion improved load balancing across MPI domains. Disabling file I/O, notably the output of large files containing charge and current densities, increased efficiency significantly to 92% of the linear scaling value for N = 4096 processors.

Weak scaling efficiencies exceeded 100% for N = 16 to N = 1024 processors compared to an N = 1 reference, with a maximum of 114% for N = 1024.

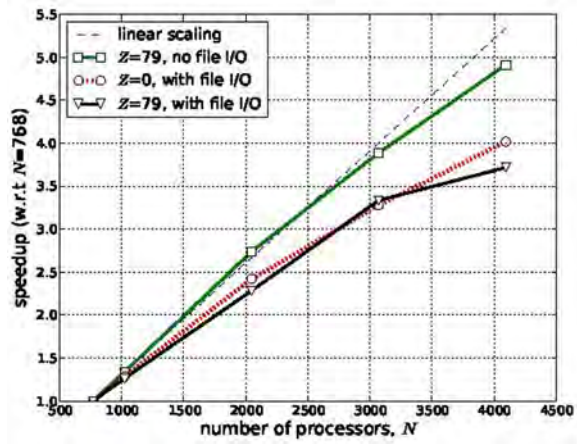


Figure 7. Strong scaling of λf PIC simulations in VORPAL on Franklin, the NERSC Cray XT4 massively parallel processing (MPP) system.

6. Conclusions

VORPAL λf PIC simulations accurately model Debye shielding of both stationary and moving single ions in isotropic and anisotropic plasmas. The shielding response to multiple ions is linear. Since these simulations scale efficiently up to thousands of processors, it will be possible to model shielding of many ions, and hence characterize the modulator section of a CeC system. Initial FEL simulations suggest that the small electron density perturbations associated with Debye shielding can be amplified to create the electric field profiles required reduce relative ion velocities. Parallel computing will enable end-to-end simulations of future CeC experiments.

Acknowledgments

This work is supported by the US DOE Office of Science, Office of Nuclear Physics, in part through the SciDAC project ComPASS and in part through the SBIR program, under contracts DE-FC02-07ER41499 and DE-FG02-08ER85182. We used resources of the National Energy Research Scientific Computing Center, DOE contract No. DE-AC02-05CH11231. The authors would like to thank R. Soliday at Argonne National Laboratory for assistance with GENESIS and using the SDDS Toolkit. The authors would also like to thank the VORPAL development team, and members of the Accelerator R&D Division at Brookhaven National Laboratory.

References

[1] G. Bell, D. Bruhwiler, A. Fedotov, A. Sobol, R. Busby, P. Stoltz, D. Abell, P. Messmer, I. Ben-Zvi, and V. Litvinenko. Simulating the dynamical friction force on ions due to a briefly co-propagating electron beam. *J. Comput. Phys.*, 227(19):8714–8735, 2008.

[2] L.-N. Hau and W.-Z. Fu. Mathematical and physical aspects of kappa velocity distribution. *Phys. Plasmas*, 14(11):110702, 2007.

[3] M.R. Jalal and F. Aghamir. Temporal and spatial evolution of electron bunching in a one-dimensional pulse. *Opt. Commun.*, 281(14):3771–3775, 2008.

[4] V.N. Litvinenko and Y.S. Derbenev. Coherent electron cooling. *Phys. Rev. Lett.*, 102(11):114801, 2009.

[5] B.W.J. McNeil and G.R.M. Robb. Correction and simplification of: `bunching from a pulse of particles obeying poisson statistics'. *J. Phys. D*, 31(4):371, 1998.
[6] D.R. Nicholson. *Introduction to Plasma Theory*. John Wiley & Sons, 1983.
[7] S. Reiche. *GENESIS 1.3 User Manual*, 2004. http://pbpl.physics.ucla.edu/_reiche.
[8] G. Wang and M. Blaskiewicz. Dynamics of ion shielding in an anisotropic electron plasma. *Phys. Rev. E*, 78(2):026413, 2008.
[9] N. Xiang, J.R. Cary, D.C. Barnes, and J. Carlsson. Low-noise electromagnetic delta f particle-in-cell simulation of electron bernstein waves. *Phys. Plasmas*, 13(6):062111, 2006.

Recent progress in accelerator physics simulations with the VORPAL code

P.H. Stoltz,[1] S.A. Veitzer,[1] C. Nieter,[1] J.R. Cary,[1] K. Amyx,[1] P. Messmer,[1] P. LeBrun,[2] P. Spentzouris,[2] J.F. Amundson,[2] Hong-Jun Kim,[3] and Tim Tautges[3]

[1]Tech-X Corporation, 5621 Arapahoe Ave., Suite A, Boulder, CO, 80303
[2]Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510-5011
[3]Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439

E-mail: pstoltz@txcorp.com

Abstract. We discuss three recent accomplishments with the VORPAL code: (i) electron cloud modeling of the Fermilab Main Injector, (ii) EBMesh, a new embedded boundary mesh generation tool for finite difference grids, and (iii) a first implementation on a graphics processing unit of the particle charge deposition algorithm for the electrostatic solver. For the Main Injector electron cloud modeling, we show results for the rate of cloud accumulation and saturation as a function of the secondary electron yield curve. Particularly we show that for a secondary electron yield curve with a peak of 2.0 and above, the cloud reaches saturation in a time shorter than the beam revolution time. For the grid generation capability, we show results meshing a nine-cell Tesla-style cavity. For the graphics processing unit charge deposition algorithm, we compare two techniques: a segmented sort and an atomic add. We show the atomic add is advantageous for simulations with fewer particles per cell, but segmented sort is advantageous for simulations with more particles per cell because of thread contention in the atomic add algorithm. Tech-X Corporation, 5621 Arapahoe Ave., Suite A, Boulder, CO, 80303

1. Electron Cloud Modeling

Electron clouds are expected to affect the design of future high-brightness proton beam accelerators, such as the proposed Project X at Fermilab and other accelerators to be used as muon sources and neutrino factories. The buildup of electron clouds is a complex interaction between the material properties of the beam pipe, the positive potential of the proton beam, and the configuration of magnetic fields. Electrons can build up rapidly when initial seed electrons, caused by residual gas ionization, are accelerated by the beam potential and impact the beam pipe walls. The cloud density will typically saturate when new secondary electrons are sufficiently shielded from the beam potential by the electron cloud. One promising mitigation technique is to coat beam pipe walls with materials that yield fewer secondary electrons. Secondary electron yield curves are energy dependent, but a main factor in terms of cloud buildup is the maximum of the secondary electron yield (SEY) curve. We have performed simulations with the VORPAL code [Nieter04] to assess the effect of different SEY curves on the buildup and time to saturation of electron clouds in the Main Injector. For these simulations we considered the buildup in a dipole section of the accelerator, with two different models, and with different curves for the SEY.

Figure 1 shows simulated electron cloud buildup for the different SEY models. In the cases of the Furman-Pivi model [Furman2002] with a maximum SEY of 2.0 (red), and the Vaughan model [Vaughan] with a maximum SEY of 3.6 (purple), the electrons build up to saturation quickly, in about 300 ns (just over 1/4 of the Main Injector revolution period). In the case of the Vaughan model with maximum SEY of 1.8 (green), the growth is slower and does not reach saturation in the course of our simulations. In the case of the Vaughan model and a maximum of 1.1 (blue), no growth is observed. These numbers are consistent with the results of Furman [Furman], who also modeled the Main Injector for a slightly different beampipe configuration (Furman determined a maximum SEY of 1.4 was the critical value). The fact that time to saturation is about the same for the cases where the maximum SEY are 2.0 and 3.6 may reflect that above a certain value, the buildup is limited by the space charge of the cloud itself, which partially screens the beam potential. We have reported elsewhere further details on the dynamics of the electron cloud [Lebrun], including details on the electromagnetic potentials and plasma wave frequencies.

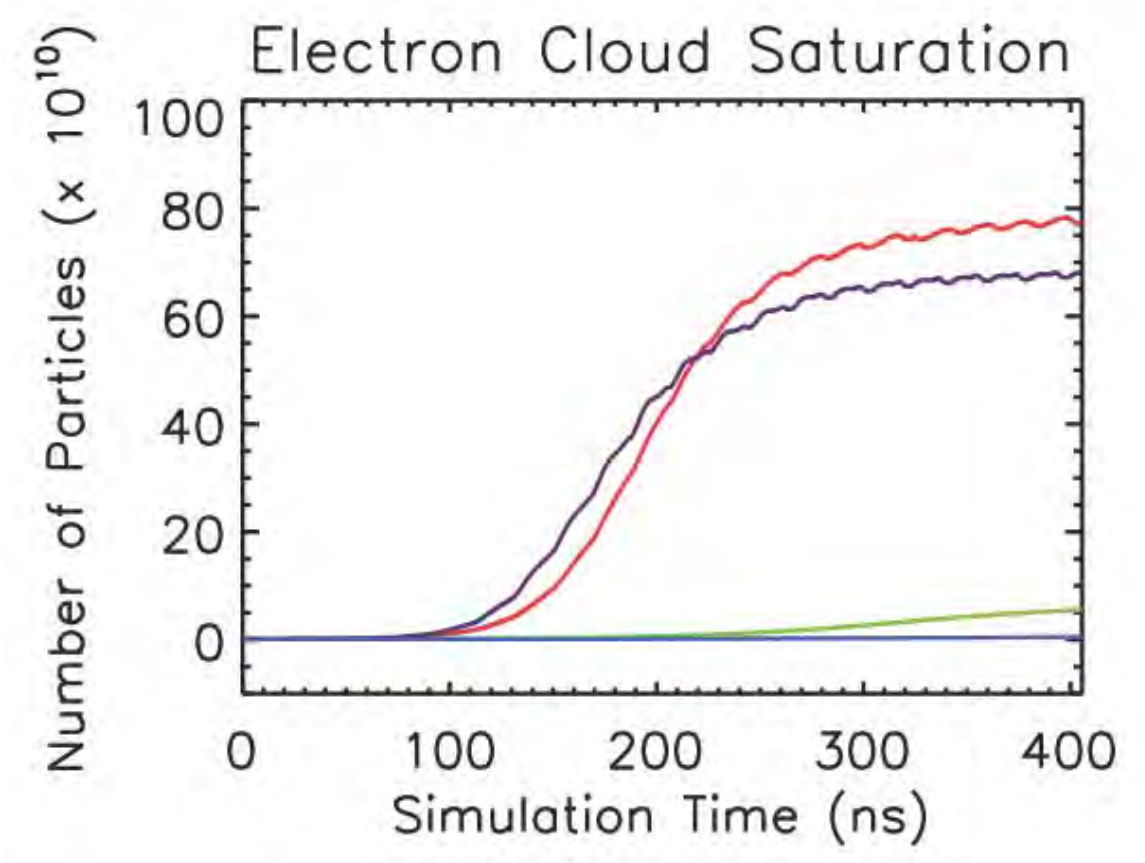


Figure 1. Electron cloud buildup for different SEY models.

2. Improved Grid Generation

A challenge for finite-difference modeling is how best to represent complex curved geometries on a Cartesian grid. A typical scenario is that researchers generate the geometry representing an accelerating cavity (see the left-hand side of Figure 2 for example of such a cavity) using a computer-aided design (CAD) tool, and the resulting file describing the geometry is in a CAD format. To embed that complex geometry into a Cartesian mesh, one needs a rapid and robust algorithm for computing information about the intersections of the Cartesian mesh with the object. For VORPAL, the

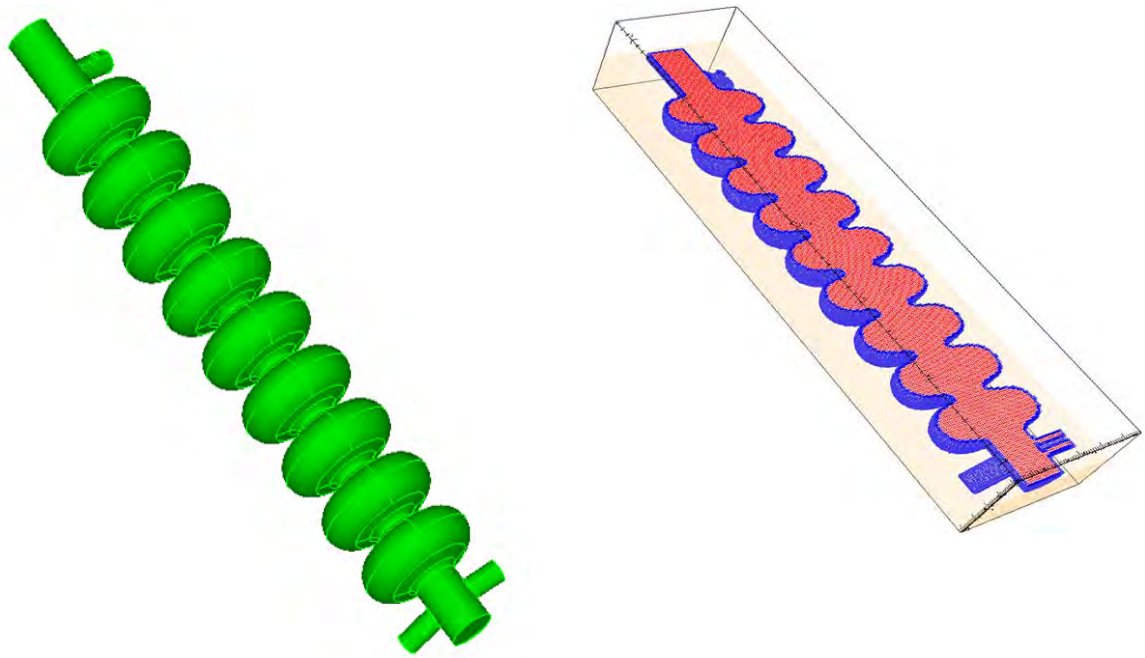


Figure 2. An example of a geometry imported by EBMesh (left) and the Cartesian mesh created (right). The geometry is a nine-cell Tesla-style accelerating cavity. The resulting Cartesian mesh had 10.7 million cells (roughly 120 × 120 × 740). The meshing operation took 20 s in serial on the 3 GHz Xeon CPU.

information required is not just which cells intersect the object, but also the side lengths from the grid corners to the intersection points, the areas of the grid surfaces formed by those cut sides, and the volumes interior to the object in the intersected cells.

To address these challenges, researchers at Argonne National Laboratory are developing the EBMesh tool [Hjkim], which uses a ray-tracing technique to increase mesh generation performance. In EBMesh, each mesh cell is distinguished as being inside, outside, or on the boundary of the input geometry. This is determined by firing rays parallel to each of the three axes. The most computationally expensive part of the embedded boundary mesh generation, the edge-geometry intersection test, is performed for the group of edges on a fired ray line together. This decreases the computational complexity of the method significantly, achieving $CN^2 \log N_t$ scaling, where other algorithms are typically $CN^3 \log N_t$ (Nt is the number of triangles on the boundary and N is the number of cells in each direction of the Cartesian mesh). In addition to the list of boundary cells, EBMesh also produces edge-cut fraction information and volume cut fraction information for each boundary cell. EBMesh can directly import various CAD-based solid model formats such as IGES, STEP and STL, and can export in a variety of formats as well. However, to avoid the overhead of interacting through files, EBMesh is linked as a library directly to VORPAL.

We show in Figure 2 an example of a geometry imported by EBMesh (left) and the Cartesian mesh created (right). In this case, the geometry is a nine-cell Tesla-style accelerating cavity. The resulting Cartesian mesh had 10.7 million cells (roughly 120 × 120 × 740). The meshing operation took 20 s in serial on the 3 GHz Xeon CPU, excluding the time to import the geometry (importing the geometry took an additional 48 s).

3. Electrostatic Charge Deposition on the GPU

A main challenge in developing electrostatic PIC algorithms for the GPU is the charge deposition stage, in which particles update the eight nearest values of the deposition field (typically the eight corners of the grid cell in which they are located). Nearby particles must update the same field values,

which requires careful handling of race conditions on the GPU. We have compared two solutions to this problem, a segmented scan algorithm and an atomic add algorithm (see Figure 3).

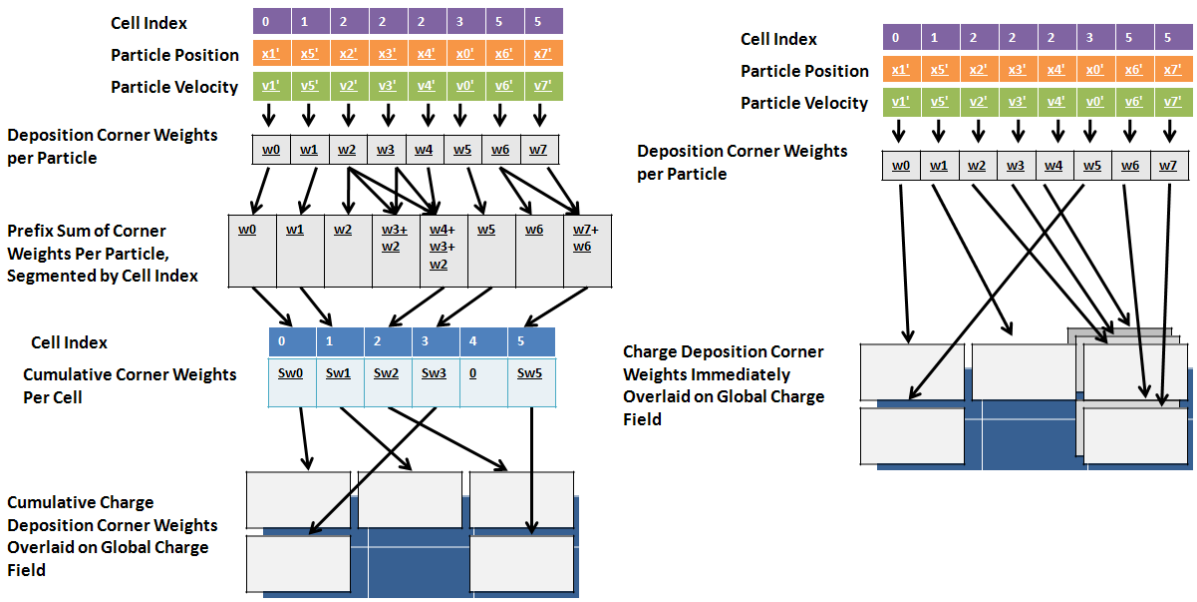


Figure 3. Illustration of segmented scan and atomic add charge deposition algorithms. On the left, the charge deposition corner weights (eight values in three dimensions) are calculated from sorted particle positions. A prefix sum, segmented by cell index, is performed to calculate the cumulative corner weights of multiple particles per cell. Finally, the resulting per cell weights are applied to the field, one corner at a time to eliminate race conditions. On the right, the charge deposition corner weights are applied directly to the charge deposition field via atomic add operations, significantly simplifying the algorithm at the cost of introducing serialized memory accesses. This algorithm is only efficient on newer Fermi-architecture GPUs.

In the first method, we perform a keyed sort (by cell index) on the particles every time step. Each particle then writes its eight corner deposition weights to linear arrays, and we perform a segmented prefix sum operation (segmented by cell index) on these arrays to calculate the cumulative deposition weights in each cell. The global field is then updated one corner at a time with the values from the scanned deposition weight vectors, thereby avoiding any race conditions. Segmented scans on a GPU are of order (N/P)log(N/P) for vectors of length N and number of processors P (modern GPUs have a value of P in the hundreds). Thus, a segmented scan parallelizes an otherwise serial operation. Segmented prefix sum algorithms are widely-used in GPU programming [GPU], and we use the implementation available in the open-source Cuda Data Parallel Primitives Library (CUDPP). The keyed sort we use is also taken from the CUDPP library.

In the second method, we perform an atomic add, in which multiple threads update the same value in memory in a thread-safe manner. The atomic add has only recently become feasible with the release of the new NVIDIA Fermi™ architecture of GPUs, which adds some memory caching. Caching allows updating the floating-point value of a specific location in memory in a thread-safe manner. This capability then allows us to perform the charge deposition step in a single operation. Although atomic operations are typically inefficient due to these serialized memory accesses, parallel charge deposition algorithms that completely avoid thread contention (as in the segmented scan algorithm described above) are so complex and require so much memory bandwidth that an algorithm that deposits charge via the serialized (but cached) atomic updates can yield equivalent or better performance in many

cases. Furthermore, using atomic operations significantly reduces the lines of code required for such an algorithm (in our case by nearly an order of magnitude).

Figure 4 shows the performance of these two algorithms compared to a CPU implementation for varying numbers of particles per cell on a three-dimensional grid with twenty cells per direction. Due to less code overhead, the atomic add algorithm outperforms the segmented scan for small numbers of particles per cell. However, for large numbers of particles per cell, the performance of the atomic deposition is nearly independent of number of particles (the performance at 10 particles per cell and 100 particles per cell only differ by ten percent). This is expected, as the atomic add algorithm does result in competition to write to the same memory location, and this competition gets worse as there are more particles in a cell.

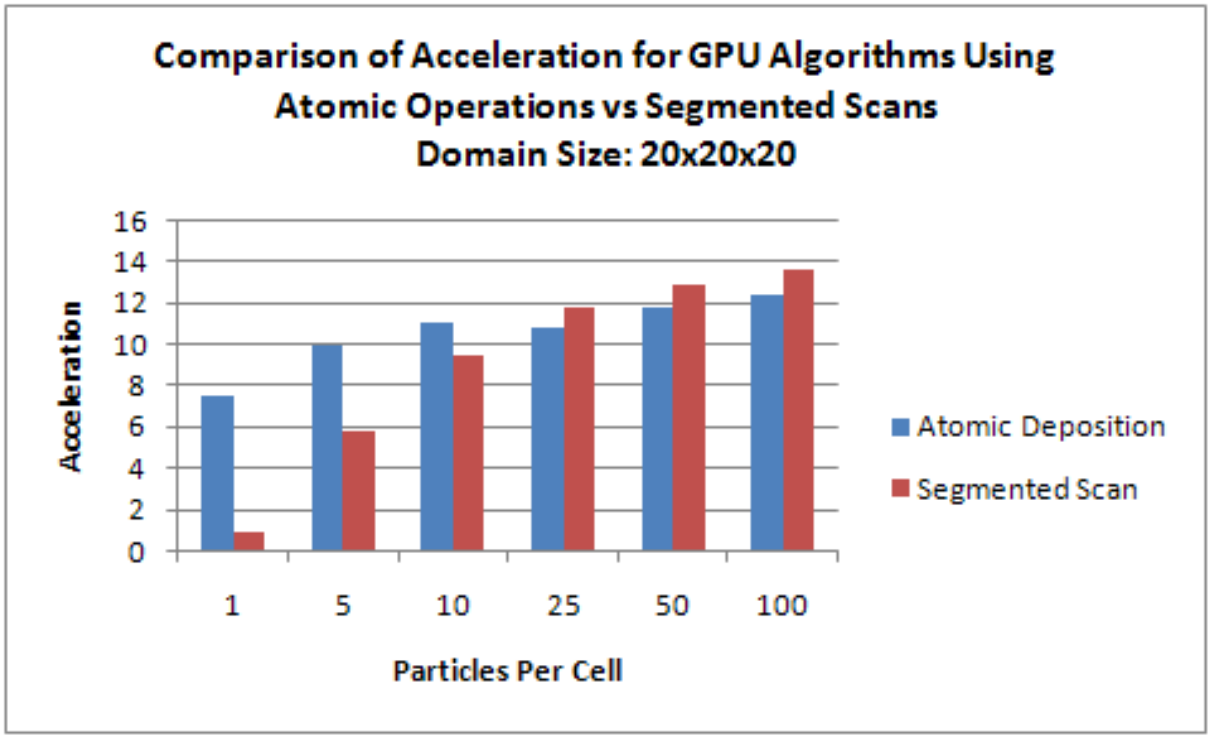


Figure 4. Comparison of performance of a GPU charge deposition algorithm based on the segmented scan and atomic add algorithms relative to a CPU implementation for varying numbers of particles per cell. Due to less code overhead, the atomic add algorithm outperforms the segmented scan for small numbers of particles per cell. For large numbers of particles per cell, the performance of the atomic deposition is nearly independent of number of particles because it does cause competition to write to the same memory location.

Acknowledgments
Part of this work was performed as part of the ComPASS SciDAC grant from the Department of Energy (grant \#DE-FC02-07ER41499). The authors acknowledge the VORPAL development team.

References
[Nieter04] Nieter C, and Cary JR, 2004, J. Comput. Phys., 196, pp. 448–73.
[Furman2002] MA Furman and MTF Pivi, "Probabilistic model for the simulation of secondary electron emission," Phys. Rev. ST Accel. Beams, 5, 124404 2002.
[Vaughan] J.R.M. Vaughan, Multipactor, IEEE Trans. Electron Devices 35 1172, 1988.
[Furman] M. A. Furman, "HINS R&D Collaboration on Electron Cloud Effects: Electron-Cloud Build-Up in the FNAL Main Injector," CBP Technote-738, 2006.

[Lebrun] P. Lebrun, P. Spentouriz, J. Cary, P. Stoltz, and S. Veitzer, "Accurate Simulation of the Electron Cloud in the Fermilab Main Injector with VORPAL," IPAC 2010, Kyoto, Japan, http://cern.ch/AccelConf/IPAC10/papers/tupd015.pdf.

[Hjkim] HJ Kim and TJ Tautges, EBMesh: An Embedded Boundary Meshing Tool, 19th International Meshing Roundtable, in preparation.

[GPU] S. Sengupta, Mark Harris, Yao Zhang, and J.D. Owens "Scan Primitives for GPU computing," Graphics Hardware, pages 97–106, 2007.

Advanced software packages for beam dynamics simulation*

J. Xu, P.N. Ostroumov, P. Fischer, B. Mustapha, M. Min, and J. Nolen

Argonne National Laboratory, Argonne, IL 60439, USA

E-mail: jin_xu@anl.gov

Abstract: We are developing highly scalable and systematic software packages for beam dynamics simulations. Both traditional Particle-In-Cell (PIC) methods and advanced direct VLASOV methods have been adopted. For the PIC method, particles are distributed evenly on different processors and space charge effects are included by solving Poisson's equation on a finite mesh. Several Poisson solvers have been developed using Fourier, hp-Finite Element (hp-FEM) and wavelet methods. These solvers can be used in Cartesian and Cylindrical coordinate systems, hp-FEM based solvers can be used on both structured and unstructured meshes. Domain decomposition (DD) has been used to parallelize these solvers and all these solvers have been implemented into the PTRACK code. PTRACK is now widely used for large scale beam dynamics simulations in linear accelerators. For the direct VLASOV method, Semi-LAGARANGE and discontinuous GALERKIN methods scheme have been employed to solve VLASOV equation directly in 1P1V and 2P2V phase spaces. They all use the time splitting scheme proposed by Cheng and Norr. Simulation results will be presented and challenges will be discussed.

1. Introduction

Plasma and charged beams are of great importance in modern science and technology. The research in this field is relying more and more on simulations. Simulating plasma and charged beams has three different methods: microscopic model, kinetic model and fluid model. In the microscopic model, each charged particle is described by 6 variables (x, y, z, v_x, v_y, v_z). Therefore, for N particles, there are 6N variables in total. This requires solving the VLASOV equation in 6N dimensions, which exceeds the capability of current supercomputers for large N. On the other end is the fluid model which is the simplest model, because it treats the plasma as a conducting fluid with electromagnetic forces exerted on it. This leads to solving the Magneto-hydrodynamics (MHD) equations in 3D (x, y and z). MHD solves for the average quantities, such as density and charge, which makes it difficult to describe the fine structure in the plasma. Between these two models is the kinetic model, which solves for the charge density function by solving the BOLTZMANN or VLASOV equations in 6 dimensions (x, y, z, v_x, v_y, v_z). The VLASOV equation describes the evolution of a system of particles under the effects of self-consistent electromagnetic fields. The model been used in current beam dynamics simulations is the kinetic model. There are two different ways to solve it. The dominant one is the so called Particle-In-Cell (PIC) method, which utilizes the motion of the particles along the characteristics of the VLASOV equation using a Lagrange-Euler approach [1,2]. The PIC method has the advantages of

* This work was supported by the U.S. Department of Energy, Office of Nuclear Physics, under Contract No. DE-AC02-06CH11357.

speed and easy implementation, but similar to MHD, it is hard to capture the fine structures in the plasma. Furthermore, there is noise associated with the finite number of particles in the simulations. With petascale computing, one-to-one simulation can be realized for 10^9 particles. But for more intense beams, PIC method still uses macro particles. The other way to solve the kinetic model is to solve the VLASOV equation directly. But the challenge is the high dimensions. For example, in order to simulate a beam in 3D, VLASOV equation has to be solved in 6D. This clearly needs peta-scale computing, and peta-scalable algorithms are critical to the success. During the past 5 years, we have developed several software packages to meet these demands. This paper presents our efforts on developing peta-scalable algorithms to simulate charged beams in linear accelerators.

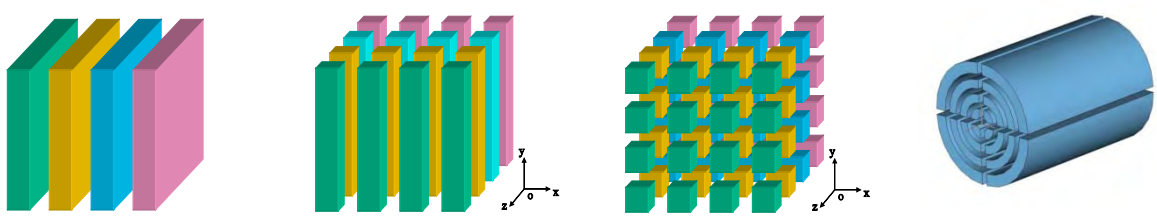
2. Parallel Poisson Solvers

In both PIC and VLASOV approaches, Poisson's equation has to be solved in 2D or 3D to account for the space charge effect. In designing peta-scalable algorithms, it is usually the most challenging part. Therefore, we present our work on this first. Several different methods have been adopted and are given in following:

$$\phi(x,y,z,t) = \sum_{m=-M/2}^{M/2-1} \sum_{p=-P/2}^{P/2-1} \sum_{n=-N/2}^{N/2-1} \phi(m,p,n,t) e^{-i\alpha m x} e^{-i\beta p y} e^{-i\gamma n z}$$

2.1. Fourier Method

This is the most standard method for solving the Poisson's equation in box region with Cartesian coordinate system. The potential has been expanded in Fourier series in all three directions. Periodic and Dirichlet zero boundary conditions have been applied in all three directions. Three different domain decomposition methods have been implemented as shown in Figure 1. Using model C, it is easy to use tens of thousands of processors with relatively small grid for space charge calculation. For example, solving the Poisson's equation on 32^3 mesh can use 32 thousands processors with model C, while only one thousand with model B and 32 processors with model A. This makes it possible to use small mesh for the space charge calculation. Since relatively small grid can be used for space charge calculation, good scaling has been obtained and can be found in [3,4].



Model A **Model B** **Model C** **Cylinder**

Figure 1. Domain decomposition method for solving Poisson's equation in Cartesian (1, 2 and 3) and Cylinder (4) coordinate systems.

2.2. Fourier Hp-Finite Element Method

This solver is developed for cylindrical coordinate system. The potential is expanded in Fourier series in the axial and circumferential directions, while it uses hp-finite element expansion in the radial direction.

$$\phi(x,y,z,t) = \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} \sum_{0}^{P} \phi(m,p,n,t) e^{-i\alpha m x} e^{-i\beta n z} P_p(y) \qquad P_p(y) = \begin{cases} (\frac{1-y}{2}) & p=0 \\ (\frac{1-y}{2})(\frac{1+y}{2}) P_{p-1}^{1,1}(y) & 0<p<P \\ (\frac{1+y}{2}) & p=P \end{cases}$$

Domain decomposition in the radial and circumferential directions has been implemented as shown in Figure 1(4). Periodic B.C. has been applied in the axial and circumferential directions and zero Dirichlet B.C. have been applied in the cylinder wall. Detailed method and benchmark results can be found in [2].

2.3. Hp-Finite Element Method on Structured Grid

The hp-Finite Element Method (hp-FEM), originated in the 1940's [6,7,8], has been applied in many different areas. In our VLASOV solvers, a parallel Poisson solver based on hp-FEM on structured grid has been constructed. 2D bases have been shown on the left of the Figure 2(1). The 2D structured mesh is shown in Figure 2(2). Continuous Galerkin (CG) method has been used and zero Dirichlet B.C. has been imposed. The potential distribution is shown in Figure 2(3). Due to the memory limitation, only the iterative solver can be used for solving boundary modes of the 2D Poisson's equation when the mesh is large. Interior modes in each element have been solved directly according to the Shur complement. The discrete system of Poisson's equation can be written as: (b and i correspond to boundary and interior variables)

$$(A_{bb} - C_{bi}A_{ii}^{-1}C_{bi}^{T})u_b = f_b - C_{bi}A_{ii}^{-1}f_i \qquad \begin{pmatrix} A_{bb} & C_{bi} \\ C_{bi}^{T} & A_{ii} \end{pmatrix}\begin{pmatrix} u_b \\ u_i \end{pmatrix} = \begin{pmatrix} f_b \\ f_i \end{pmatrix}$$

$$u_i = A_{ii}^{-1}(f_i - C_{bi}^{T}u_b)$$

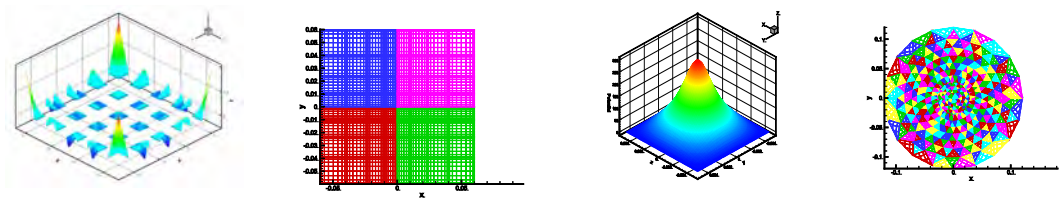


Figure 2. Convergence history with/out preconditioning (left), wavelet coefficients (middle) and 2D structured grid (right).

2.4. Hp-Finite Element Method on Unstructured Grid

In order to solve Poisson's equation in complex geometries, a parallel Poisson solver using hp-FEM on an unstructured grid has been developed recently. Since finite element method (FEM) can handle complex geometry easily, and spectral method can achieve high order accuracy. Combine these two, hp-FEM can handle the complex geometry and also achieve high order accuracy at the same time. The 2D unstructured mesh is shown in Figure 2(4).

The potential can be expressed as

$$f(r,s) = \sum_{i+j\le p} \psi_{ij}(r,s)\hat{f}_{ij}$$

The continuous Galerkin formula for solving the Poisson's equation is

$$\begin{aligned}
\int_V f\cdot\varphi dv &= \int_V \nabla^2 u\cdot\varphi dv \\
&= \int_V [\nabla(\nabla u\varphi) - \nabla u\cdot\nabla\varphi]dv \\
&= \int_{\partial V} \vec{n}\cdot(\nabla u\varphi)dS - \int_V \nabla u\cdot\nabla\varphi dv \\
\int_V \nabla u\cdot\nabla\varphi dv &= \int_{\partial V} \vec{n}\cdot(\nabla u\varphi)dS - \int_V f\cdot\varphi dv \\
\int_{\partial V} \vec{n}\cdot(\nabla u\varphi)dS &= \int_{\partial V} (n_x\varphi\frac{\partial u}{\partial x} + n_y\varphi\frac{\partial u}{\partial y})dS
\end{aligned}$$

3. Beam Dynamics Simulation With PIC Method

In the last several years, we have developed a parallel PIC solver based on the serial version of TRACK, which was developed in the physics division at ANL. Particles have been distributed evenly over all processors, and parallel Poisson solvers described above have been used for the space charge effect. Parallel algorithm and detailed benchmark results can be found in [2,3,4]. As small mesh can be used for calculating space charge effect, PTRACK has achieved good scaling. Recently PTRACK has been used for end-to-end simulation of full LINAC system. It can also be used for one-to-one simulation for some beams such as that in FNAL proton driver. Totally 865M charged particles have been simulated from 50 keV to 2.5 MeV in 325 MHz radio frequency quadrupole of a proton driver at FNAL. Figure 3 shows the comparison of contour in ($\phi, \Delta W/W$) plane with 1M, 10M, 100M and 865M particles. As can be seen, using large number of particles provides much more accurate information and this is useful to the accelerator design and optimizations. Now PTRACK is being used as workhorse for large scale optimizations.

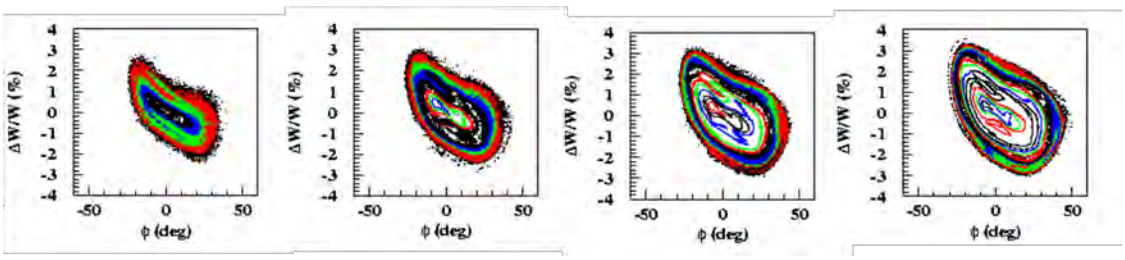


Figure 3. ($\phi, \Delta W/W$) plane contour with 1M(1), 10M(2), 100M(3) and 865M(4) particles.

4. Beam Dynamics Simulation By Direct Solving Vlasov Equation

In order to overcome the shortcoming of the PIC solvers, we have developed direct VLASOV solvers. The distribution function $f(\vec{x},\vec{v},t)$ in phase space is governed by the VLASOV equation.

VLASOV equation in 1P1V phase space

$$\frac{\partial f(x,v,t)}{\partial t} + v(x,t)\frac{\partial f(x,v,t)}{\partial x} + E(x,t)\frac{\partial f(x,v,t)}{\partial v} = 0, \quad E(x,t) = -\frac{\partial \phi(x,t)}{\partial x}, -\Delta\phi(x,t) = \frac{\partial E(x,t)}{\partial x} = \rho(x,t) - 1$$

4.1. VLASOV Equation in 2P2V Phase Space

$$\frac{\partial f}{\partial z} + \frac{\vec{v}}{v_b}\cdot\nabla_{\vec{x}} f + \frac{q}{\gamma_b m v_b}\left(-\frac{1}{\gamma_b^2}\nabla\Phi^s + \vec{E}^e + (\vec{v},v_b)^T \times \vec{B}^e\right)\cdot\nabla_{\vec{v}} f = 0, \quad -\Delta_{\vec{x}}\Phi^s = \frac{q}{\varepsilon_0}\int_{R^2} f(z,\vec{x},\vec{v})d\vec{v}$$

where Φ^s is the self-consistent electric potential due to charges. $\vec{E}^e$ and $\vec{B}^e$ are external electric and magnetic fields. v_b is the reference beam velocity.

4.2. Numerical Algorithm

The Semi-Lagrangian Method (SLM) [9] has been used for time integration. A plot explains the idea is shown on the left of Figure 4(1). The time splitting scheme has been used for time integration as proposed by Cheng and Knorr [10]. 4D domain decomposition has been adopted as shown in Figure 4(2).

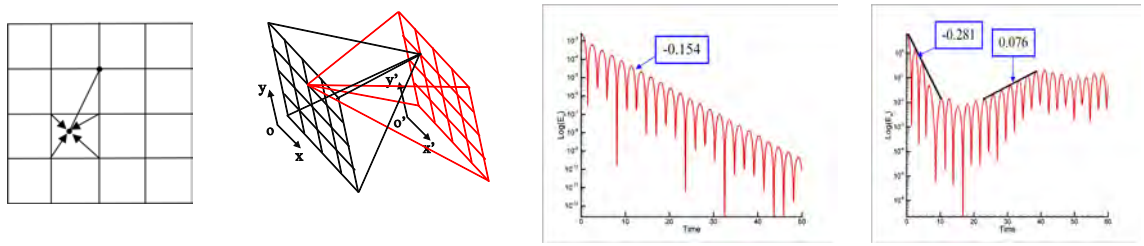


Figure 4. Semi-Lagrange Scheme (1), 4D domain decomposition (2), Linear Landau Damping (3), Strong Landau Damping (4).

4.3. Benchmarks and Simulation Results

The code comprises two major parts: interpolation and space charge (SC) calculation. The SLM performs back tracking and interpolation respectively in the physical and velocity spaces. Each processor has only part of the global mesh for the space charge calculations. The field mesh and space charge mesh are different. This scheme has the advantage of easy implementation and no communication for particle tracking is required. However, this method requires large memory in each processor and intense communication for the parallel Poisson solver.

Figures (3 and 4) in Figure 4 show the time history of log(Ex) for linear and strong Landau damping. The initial particle distribution function and the related parameters are shown in following:

$$f(0,x,v) = \frac{1}{\sqrt{2\pi}} exp(-v^2/2)(1+\alpha\, cos(kx)) \qquad L = 4\pi, \quad R = [-6,6], \quad \Delta t = 1/8,$$

$$\forall (x,v) \in [0,L] \times R, \quad \alpha = 0.01, \quad k = 0.5 \qquad P = 16, \quad E = 64, \quad 1024 \times 1024 \quad CPU = 256, \quad T \sim 10\ mins$$

For the linear Landau damping, alpha=0.01, and for the strong Landau damping, alpha is 0.5. Clearly they represent different dynamics. The decreasing and increasing rate can be measured and are consistent with theoretical predictions and the work by other researchers.

4.4. 2P2V Simulations

In 2P2V simulations, a proton beam has been simulated through alternating hard edge electric quadruple channel. The initial emittance is $\varepsilon = 200\pi$ mm mrad, and the energy is W=0.2 MeV. The current of the beam is 0.1 A, and the reference velocity is $v_b = 6.19 \times 10^6$ m/s. The transverse physical space is [–0.12, 0.12] by [–0.12, 0.12], and the velocity space is [-8×10^5, 8×10^5] by [-8×10^5, 8×10^5] m/s. The alternating electric quadruple field is defined as $\vec{E}^e(x,y,z) = (k_0(z)x, -k_0(z)y)$. Figure 5 compare with PIC simulations, and Figure 6 are contours.

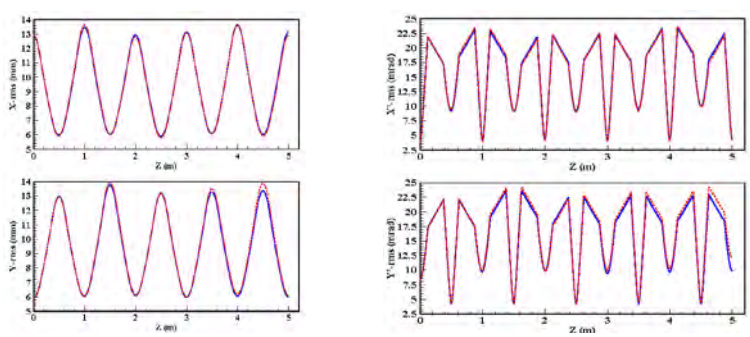


Figure 5. Comparison of TRACK (solid) and VLASOV (dotted) simulations using a Gaussian beam: *Xrms* (*left upper)*, *Yrms* (*lower left*), *X′rms* (*upper right*), *Yr′ms* (*lower right*) for a 100 mA proton beam.

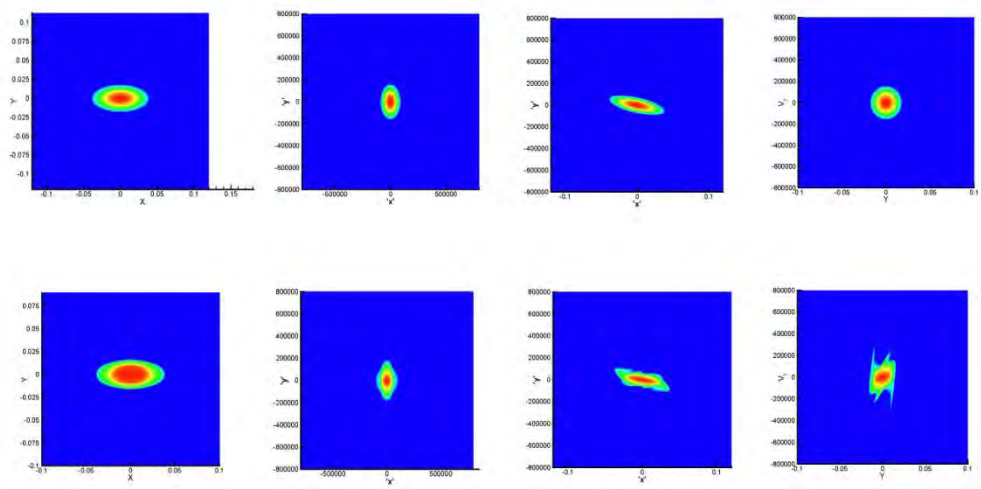


Figure 6. From left to right are contours in the (x, y), (x, x'), (y, y') and (x', y') planes, from top to bottom correspond to t=0 and 3 time period.

5. Summary

This paper presents our researches on developing peta-scalable algorithms for large scale beam dynamics simulations. Both PIC method and direct VLASOV method have been used. Different parallel Poisson solvers have been developed to satisfy the requirement of considering space charge effect in various solvers. Domain decomposition has been adopted for parallelization of the TRACK code. PTRACK has now been used for large scale beam dynamic optimization and real accelerator simulations. These Poisson solvers adopted different numerical techniques in different conditions, such as using Cartesian and Cylindrical coordinate systems, using structure and unstructured grids, etc. Direct VLASOV solvers have been developed for 2D and 4D. A high-order hp-FEM has been used. The advantages and effectiveness of the hp-FEM have been demonstrated. The VLASOV solvers have adopted the Semi-Lagrangian method. Similarly domain decomposition has been used for parallelization of these solvers. Scalable Poisson solvers have been developed with hp-FEM. Linear and strong Landau damping have been studied with direct 2D VLASOV solver, and results clearly captured the physics of these phenomena. Direct 2D VLASOV solver can be applied to study more problems in plasma and charged beams later. Benchmarks of the parallel models have shown good scaling on BlueGene/P at ANL with up to 32k processors. The hp-FEM shows its advantages in these direct VLASOV solvers, such as local interpolation, easy parallelization and long time integration. These developments are encouraging, and more investigation will be done.

Acknowledgments

We would like to thank ALCF support team at ANL. This work was completed at ANL. This work was supported by the U.S. Department of Energy, Office of Nuclear Physics, under Contract No. DE-AC02-06CH11357.

References

[1] P.N. Ostroumov, V.N. Aseev, and B. Mustapha. Phys. Rev. ST. Accel. Beams 7, 090101 (2004).
[2] J. Xu and P.N. Ostroumov, Computer Physics Communications, 178 (2008) 290–300.
[3] J. Xu, B. Mustapha, V.N. Aseev and P.N. Ostroumov, Phys. Rev. ST. Accel. Beams 10, 014201 (2007).
[4] J. Xu, B. Mustapha, V. N. Aseev and P.N. Ostroumov, HB 2008, Nashville, Tennessee, USA, Aug. 25–29, 2008.
[5] M. Gutnic, M. Haefele, I. Paun and E. Sonnendrücker, Comput. Phys. Comm., 164, 214–219 (2004).

[6] E. Sonnendrücker, J. Roche, P. Bertrand and A. Ghizzo, J. Comput. Phys., 149, 201 (1998).
[7] F. Filbet, E. Sonnendrücker, Vol.16, No.5, 763–791 (2006).
[8] M.O. Deville, P.F. Fischer and E.H. Mund, High-Order Methods for Incompressible Fluid Flow, Cambridge University Press, Cambridge (2002).
[9] J.S. Hesthaven and T. Warburton, Nodal Discontinuous Galerkin Methods: Algorithms, Analysis and Applications, Springer, New York (2008).
[8] G.E. Karniadakis and S.J. Sherwin, Spectral/hp element methods for CFD, Oxford University Press, London (1999).
[9] A. Robert, Atmos. Ocean. 19, 35 (1981).
[10] C.Z. Cheng and G. Knorr, J. Sci. Comp., 22, 330 (1976).
[11] J. Xu, P.N. Ostroumov, "Solving 2D2V VLASOV Equation with High Order Spectral Element Method," Communication in Computational Physics, 2010.

Modeling the Parallel Performance of CMA-ES

Monte Lunacek, Peter Graf, Ambarish Nag, Christopher H. Chang, Kwiseon Kim, and Michael Seibert

National Renewable Energy Laboratory, Golden CO 80401

Abstract. The Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) is an effective population based optimization algorithm that can be made more efficient by evaluating each iteration in parallel. We model the parallel efficiency of CMA-ES and use this to show that, for any objective function, there is always an optimal population size that minimizes the search time by balancing the algorithm's complexity with a population size that minimizes the number of iterations necessary to converge. This research shows that, from a practical point of view, choosing the maximum number of processors available may actually increase the wall time it takes CMA-ES to optimize a particular function. We use our model to determine the optimal number of processors for a real-world application.

1. Introduction

Some photosynthetic organisms, such as green algae, are capable of transforming light energy and carbon dioxide into fuel-related molecules. Making this process more efficient requires a fundamental understanding of the underlying dynamics of the metabolic system. One way to gain this insight is through high-performance simulation.

The High Performance Systems Biology Toolkit (*HiPer SBTK*) [1] was developed to efficiently explore and optimize large metabolic models that are constructed using a system of ordinary differential equations (ODEs). Unfortunately, the precision of the ODE solver can render gradient-based local search algorithms ineffective because of surface noise. The Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [2] is an alternative, population-based optimization algorithm that performs well in the presence of noise and does not require gradient information. Although this makes CMA-ES less efficient when compared to gradient-based methods, the efficiency of CMA-ES can be easily increased by evaluating each iteration in parallel. This make CMA-ES an attractive additional component to the HiPer SBTK.

CMA-ES uses a set of points, called a population, to determine a descent direction during each iteration. The number of iterations CMA-ES requires is impacted by the population size, which we denote λ. Increasing λ generally decreases the number of iterations required to converge [3]. At the same time, it also increases the time it takes to process each iteration. Understanding these opposing trends is critical for making informed decisions regarding the degree to which we can run CMA-ES in parallel.

Using a master-worker implementation, CMA-ES is most efficient with one master and l working processors, such that the number of processors, n, is always equal to λ+1. The parallel performance of CMA-ES is, therefore, determined by the population size, λ.

This paper analyzes the parallel performance of CMA-ES in several ways. First, we create a predictive model of CMA-ES by fitting empirical data to algebraic equations. Then we use this model to prove that there is always a value for λ that minimizes the optimization wall time, which we denote λ_o. This means that using the maximum number of processors available can lead to slower optimization

wall times. On the other hand, faster optimization doesn't necessarily utilize each processor in the most efficient way. We explore this trade-off by fitting empirical data to our model and investigating conditions where CMA-ES produces minimal optimization time and where it uses resources most efficiently. Finally, we use our model to estimate the potential speedup of CMA-ES on an example systems biology problem.

2. Related Work

Muller *et al.* introduced a parallel CMA-ES library in Fortran [4]. However, the code is not designed to run a single instance of CMA-ES on a highly parallel architecture, but rather parallel instances of multiple independent runs and cooperative island models. Other efforts have been made to reduce the algorithmic complexity of CMA-ES when run on a single computer [5]. Although this makes CMAES more efficient per iteration, it can also increase the number of iterations required to converge. Our work is unique in that we are modeling and exploring how efficient a single, highly-parallel instance of CMA-ES can be.

3. Parallel Performance Model

We use the *master-worker* paradigm to facilitate the parallel evaluation of the population. This simplifies the number of characteristics that impact the parallel performance of CMA-ES, which include the population size λ (which also indicates the number of processors used), problem dimension N, and objective function evaluation time τ. The optimization wall time is the total number of seconds CMA-ES spends optimizing a function. Given N and τ, we model the total optimization wall time, as a function of λ by decomposing it into the *number-of-iterations* required to converge, $i(\lambda)$, and the *cost-per-iteration*, $c(i)$.

$$T(\lambda) = i(\lambda) \bullet c(\lambda);$$

where $T(\lambda)$ is the total optimization wall time. Each component can be modeled using the following functional form

$$i(\lambda) = a_1 + b_1 \bullet \lambda^{-c1}$$

$$c(\lambda) = a_2 + b_2 \bullet \lambda^{c2} .$$

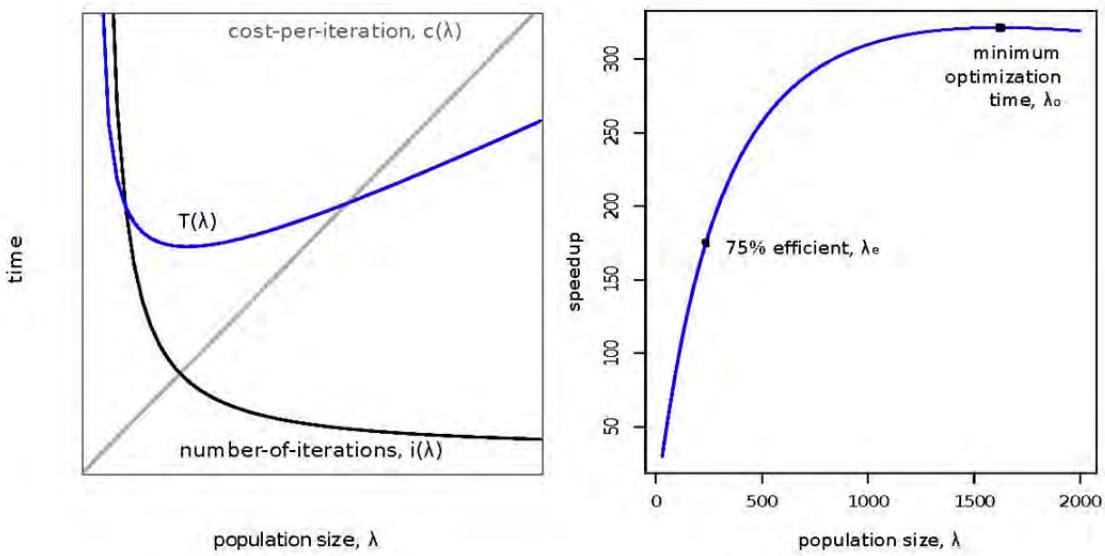
The number-of-iterations $i(\lambda)$ is problem-dependent and is not affected by the parallel implementation. The cost-per-iteration $c(\lambda)$, on the other hand, is problem-independent in that it only depends on characteristics that are easy to measure, such as N and τ, and greatly impacts the parallel performance. Figure 1(a) graphically summarizes these equations.

There's always a minimum optimization time

Given our two equations, the total optimization wall time for CMA-ES is

$$T(\lambda) = a_1a_2 + a_1b_2\lambda^{c2} + a_2b_1\lambda^{-c1} + b_1b_2\lambda^{c2} - c1 .$$

For $a1;a2;b1;b2;c1;c2 > 0$, which is a valid assumption for our models, the limit as $\lambda \to 0$ is ∞ and as $\lambda \to \infty$ is ∞. The fact that $T(\lambda) < \infty$ for all $\lambda \neq 0$ and $\lambda < \infty$ shows that $T(\lambda)$ always has a minimum.



(a) Sample model (b) Speedup, λ_o, and λ_e

Figure 1: A sample of our model (a) and the two metrics we identify to discuss performance (b): minimum optimization wall time, λ_o, and the λ value where CMA-ES is 75% efficient, λ_e. The x- and y-axis are not shown in the left graph (a) because it is only intended to show the form of our model's equations and is not based on actual data.

Trade-off: minimal optimization wall time vs. resource efficiency

We estimate $i(\lambda)$ and $c(\lambda)$ with empirical data. The cost $c(\lambda)$ is estimated by empirically sampling the three-dimensional space defined by N, λ, and τ, and running CMA-ES for 10 iterations based on these parameters. To create empirical data for $i(\lambda)$, we use an elliptical test function with a condition number of 10^9:

$$f(\vec{x}) = \sum_{i=0}^{N-1} 10^{(9 \cdot i)/(N-1)} \cdot x_i^2 .$$

We use $T(\lambda)$ to compute speedup, which is $s(n) = T(\lambda) / T(\lambda+1)$ (Figure 1 (b)) and efficiency, which is $s(n) / n$. Then we identify the λ values where CMA-ES is 75% efficient, denoted λ_e, and has the fastest optimization wall time, λ_o. We computed these metrics as a function of objective function evaluation time τ for problem dimension $N = 100$ and $N = 500$. Figure 1(b) shows these metrics on a sample speedup curve.

When the objective function is fast, lower N values produce lower λ_o values, which is shown in Figure 2(a). As the objective function evaluation time increases, there is a transition where higher N values produce lower λ_o values. Figure 2(b) displays our results for λ_e. We find that larger N produces larger λ_e values. The exception is during the transition phase, which is the gray area in Figure 2(b).

The cost-per-iteration is composed of three parts: τ, the cost of managing the MPI process, and the cost associated with the algorithm complexity of CMA-ES. Both the CMA-ES cost and the MPI cost increase with larger values of N (results not shown). This makes τ (the objective function evaluation time) a smaller percentage of the cost-per-iteration total, and helps explain why lower dimensional problems transition earlier and why they reach their λ_e values sooner. In general:

- **Slow objective function evaluation:** Lower dimensional problems can use more processors to produce minimal optimization time. Higher N values can use more processors in an efficient way.

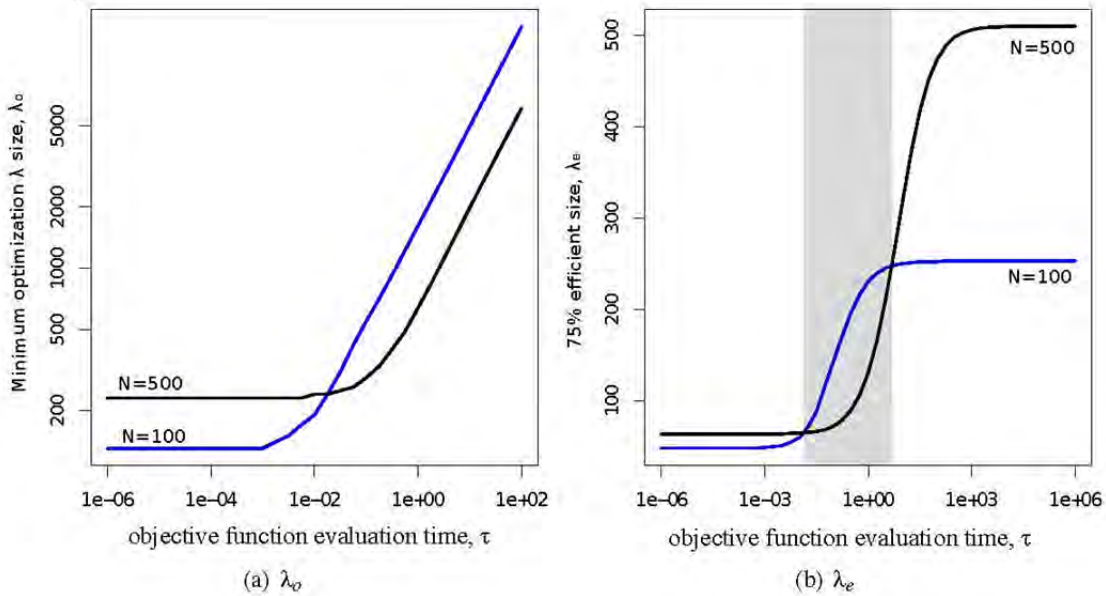


Figure 2: The λ values that produce minimum optimization wall time (a) and 75% resource efficiency (b) as a function of objective function evaluation time, τ.

- **Fast objective function evaluation:** Higher dimensional problems always produce higher λ_e and λ_o values.

4 Systems biology

As mentioned, many of the metabolic models used in the HiPer SBTK, which include our systems biology models, have noisy surfaces that can cause gradient-based optimization methods to fail. Since the objective functions are posed as a least-squares problem, the surface is also ill-conditioned. CMAES is one of the few derivative-free methods that can deal with both noisy and ill-conditioned problems.

Our model has $N = 67$ parameters and takes about $\tau = 0.15$ seconds to evaluate, on average. A typical optimization run on a single computer would take well over an hour to complete. We would like to know how much we can improve the optimization time by running this in parallel, and what the trade-offs are.

We estimate the iteration model by running CMA-ES with $\lambda = 100, 200, 400$, and 900. This can be done on a single computer or with any degree of parallelism. Then we extrapolate the cost model and generate the following information. Figure 3 explains this graphically.

metric	λ	predicted seconds
min optimization time	≈ 506	72.3
25% efficiency	≈ 226	77.5
50% efficiency	≈ 90	96.9
75% efficiency	≈ 50	118.3

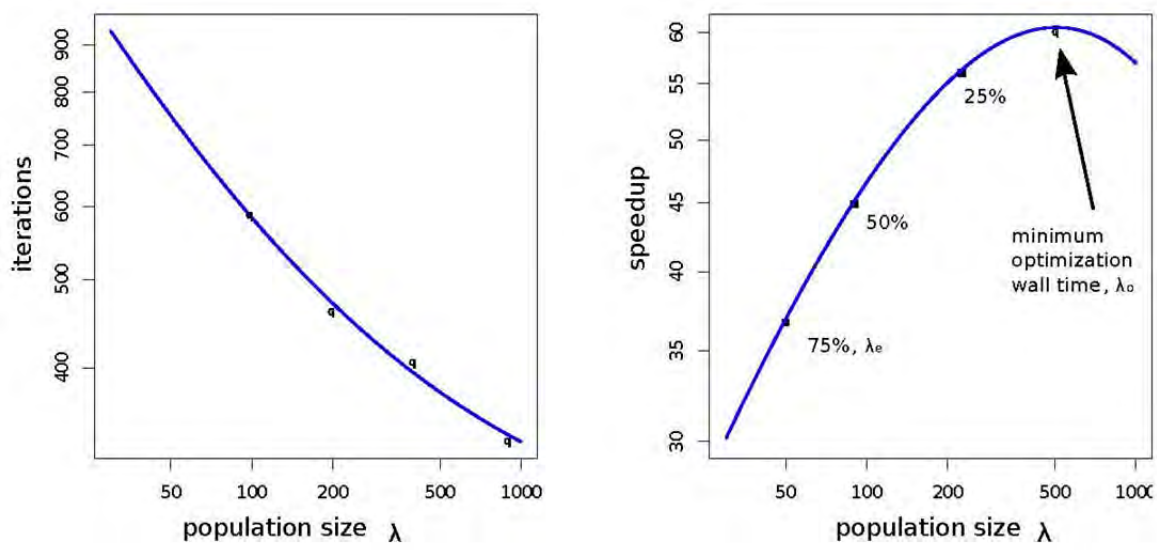


Figure 3: The estimated iteration model (left) and the speedup curve generated from our cost-periteration model. Combined, these graphs allow us to make informed decisions regarding the best population size λ.

5 Conclusions

The parallel performance of CMA-ES is composed of two parts. The cost-per-iteration model accounts for the parallel performance, and the number-of-iterations model captures problem specific characteristics. The iteration model can be estimated without any parallel considerations. The cost model is generic and can be applied to any objective function. We use this to show that there is always a λ value that minimizes optimization wall time. From a practical stand-point, we use this model to estimate potential speedup and make informed decisions with respect to the number of processors and population size.

References

[1] CH Chang, P Graf, DM Alber, K Kim, G Murray, M Posewitz, and M Seibert. Photons, photosynthesis, and high-performance computing: Challenges, progress, and promise of modelling metabolism in green algae. *J. Phys. Conf. Ser.*, 125(012048), 2008.

[2] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[3] N. Hansen, S.D. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11(1), 2003.

[4] C. L. Mueller, B. Baumgartner, G. Ofenbeck, B. Schrader, and I. F. Sbalzarini. pcmalib: a parallel fortran 90 library for the evolution strategy with covariance matrix adaptation. In *GECCO '09*: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1411–1418, New York, NY, USA, 2009. ACM.

[5] R. Ros and N. Hansen. A simple modification in cma-es achieving linear time and space complexity. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 296–305, Berlin, Heidelberg, 2008. Springer-Verlag.

Systems biology approach to discovery of phenotype-specific metabolic processes for engineering bacterial bio-hydrogen

Andrea M Rocha,[1] William Hendrix,[2,3] Matthew C Schmidt,[2,3] James R. Mihelcic,[1] Nagiza F Samatova[2, 3,*]

[1]Department of Civil & Environmental Engineering, University of South Florida, Tampa, FL 33620
[2]Department of Computer Science, North Carolina State University, Raleigh, NC 27695
[3]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831

[*]E-mail: samatovan@ornl.gov

Abstract. Biological hydrogen is one of the most promising alternative energy and fuel resource today. Due to the ability of several naturally occurring microorganisms to generate hydrogen using different metabolic processes, engineered biohydrogen production is considered a feasible and sustainable technology. To improve upon current technological applications, biological and metabolic engineering approaches are underway to optimize metabolic processes involved in hydrogen production. In order to successfully modify metabolic pathways, full understanding of metabolic networks involved in expression of microbial traits in hydrogen producing organisms is necessary. In this paper, we demonstrate the application of two computational systems biology approaches to identify phenotype-related metabolic components related to hydrogen-producing bacteria. The first method utilizes prior knowledge to discover clusters of phenotype-related proteins in individual phenotype-expressing microorganisms. When applied to bacterial functional relationship data, key enzymes, such as [FeFe]-hydrogenase, [NiFe]-hydrogenase, and glutamate synthase, were identified. In the second method, our efficient graph search algorithm, called PRP-Finder, is used to identify key genes and metabolic pathways that are likely related to biohydrogen production. In addition, predictions on a potential interplay, or cross-talk, between metabolic pathways can be predicted using this approach. Information obtained from both methodologies provides valuable insights into metabolic networks and system controls involved in expression of microbial traits essential for advancing engineering approaches that result in more efficient biohydrogen production.

1. Introduction

Production of biological hydrogen (bio-hydrogen) is potentially a feasible technology for generation of alternative energy and fuels. The ability of several naturally occurring microorganisms to generate hydrogen using various metabolic processes, such as light and dark fermentation metabolic routes, makes bio-hydrogen production a favorable option for addressing bioenergy needs (Nath and Das 2004). Unfortunately, application of bio-hydrogen technologies is often limited due to overall low hydrogen production and yields generated by hydrogen-producing microorganisms (Nath and Das 2004; Jones 2008). This is due mainly to utilization of organic substrates by microorganisms (e.g.,

glucose) towards cellular growth rather than production of hydrogen (Nath and Das 2004). To help overcome this problem, one approach being considered is the genetic modification or bioengineering of microbial metabolic pathways away from biomass production and towards expression of phenotypic traits related to hydrogen production (Jones 2008).

Genetic modification methodologies have not been widely applied towards bio-hydrogen production because a fuller understanding of metabolic processes, networks, and their relationships involved in expression of desired phenotypic traits, such as hydrogen production, is necessary. Although numerous studies have identified hydrogen-producing organisms (Nath and Das 2004), this type of information remains unclear. To close this gap, we started with developing and employing two computational systems biology approaches to identify phenotype-related metabolic processes for fermentative hydrogen production. The former, or the "bottom-up" approach called **DENSE**, capitalizes on the availability of partial "prior knowledge" about the proteins involved in this process and enriches that knowledge with newly identified sets of functionally associated proteins present in individual phenotype-expressing microorganisms. The latter, or the "top-down" approach called **PRP-Finder**, capitalizes on high-throughput comparative analysis of multiple genome-scale metabolic networks to identify phenotype-related metabolic genes and pathways. Description and application of each approach towards bio-hydrogen production is described in the following sections.

2. Bottom-Up Discovery of Phenotype-Related Proteins Using Knowledge Priors

In fermentative hydrogen-producing organisms, such as *Clostridium acetobutylicum,* hydrogen yields are dependent on the presence and activation of hydrogen producing enzymes called hydrogenases (Vignais et al. 2001). Studies evaluating the role of hydrogenase in hydrogen production have shown that organisms can contain more than one type of hydrogenase that can each require sets of accessory proteins for activation. As such, the presence or absence of specific accessory proteins plays an important role in regulating the activity of hydrogenase and hydrogen production or uptake in microorganisms. In addition, many hydrogenases are thought to either directly or indirectly regulate other metabolic processes, such as nitrogen metabolism (Rey et al. 2007). To identify phenotype-related proteins required for activation and maturation of [FeFe]-hydrogenases and understand relationships between hydrogenase and other metabolic proteins, we applied our DENSE algorithm to identify proteins that are functionally associated with the hydrogenase, HydA, in *C. acetobutylicum.*

DENSE was able to identify three maturation proteins that are essential for expression of a [FeFe]-hydrogenase (Akhtar and Jones 2008). They are HydE (CAC1631), HydF (CAC1651), and HydG (CAC1356) (Figure 1). When these proteins are present and interact with HydA1, activation of the hydrogen producing [FeFe]-hydrogenase occurs. According to studies on hydrogenases, deletion of one the proteins will result in inactivation of the [FeFe]-hydrogenase (Akhtar and Jones 2008). In addition to identifying key protein clusters, the algorithm predicted an association between an uncharacterized protein (Figure 1; CAC0487) and the three maturation proteins. Identification of uncharacterized proteins may assist scientists and bioengineers in identifying potential functional roles of unknown proteins. For example, in our study (Hendrix et al. 2010), functional associations between HydA1 and proteins involved in activation of a hydrogen producing enzyme were identified. Since the uncharacterized protein is highly interconnected with the maturation proteins, it can be predicted that the protein is involved in development of the [FeFe]-hydrogenase (HydA1). Utilizing this information, the role of

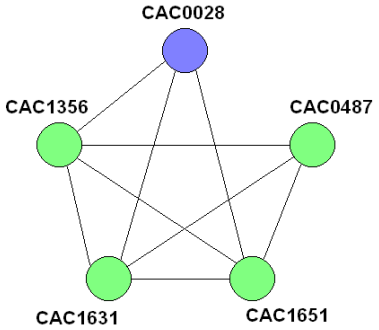


Figure 1. Dense and enriched sub-graphs identified by DENSE algorithm. Blue circle represents the knowledge prior protein HydA1. Image borrowed from Hendrix et al. (2010)

CAC0487 in relation to the three maturation proteins can be characterized through genetic studies and then applied to bioengineering hydrogen producers.

2.1. DENSE: Dense and ENnriched Sub-graph Enumeration

Given a phenotype-expressing organism, our DENSE algorithm tackles the problem of identifying genes that are functionally associated to a set of known phenotype-related proteins by enumerating the "dense and enriched" subgraphs in genome-scale networks of functionally associated or interacting proteins (Hendrix et al. 2010). We define a "dense" subgraph as one in which every vertex is adjacent to at least some γ percentage of the other vertices in the subgraph for some value γ above 50%, which correspond to a set of genes with many strong functional associations between them. In order to incorporate researchers' *prior* knowledge, though, we extend this definition by introducing an "enriched" dense subgraph in which at least μ percentage of the vertices are contained in the knowledge prior query set, for some parameter value μ. Genes contained in such dense and enriched subgraphs, or "μ-enriched, γ-dense quasi-cliques," have strong functional relationships with the previously identified genes, and so are likely to perform a related task.

DENSE uses an agglomerative approach to discover these μ-enriched, γ-dense quasi-cliques, starting with a single query vertex v_0, adding one vertex to the quasi-clique at a time, and backtracking as it finds maximal enriched quasi-cliques or subgraphs that cannot be contained in an enriched quasi-clique. Based on our theoretical results, we limit the vertices that can be added to the subgraph in order to reduce the necessary search space. A number of heuristic values are maintained, and when any of the heuristic values is reduced to zero, the related vertex becomes a "dead end" that is not added to the subgraph.

3. Top-Down High-Throughput Discovery of Phenotype-Related Metabolic Pathways

To discover phenotype-related enzymes and metabolic pathways related to bio-hydrogen production, we applied our PRP-Finder algorithm to evaluate 14 hydrogen-producing and 11 non-hydrogen producing bacteria. We selected microorganisms representative of the three different types of bio-hydrogen production—biophotolysis, light fermentation, and dark fermentation.

Our analysis of identified enzymes and reactions for hydrogen-producing organisms indicates that a large number of reactions involved in glycolysis are conserved across hydrogen producing organisms but not conserved across non-hydrogen producing organisms (Jones 2008). In addition, we identified key metabolic reactions and enzymes related to hydrogen production during mixed acid and dark fermentation (Figure 2). Specifically, we were able to identify formate hydrogen lyase (FHL), an enzyme responsible for the oxidation of formate and production of CO_2 (g) and H_2 (g) (Jones 2008).

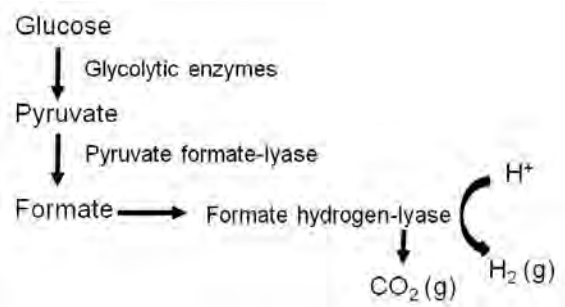


Figure 2. General overview of hydrogen production through mixed acid fermentation.

In addition to dark fermentation routes, the PRP-finder also identified metabolic reactions associated with nitrogen metabolism. These reactions include nitrogen fixation by nitrogenase, and assimilation of ammonia by glutamine synthetase and glutamate synthase. During nitrogen-fixation, N_2 (g) is converted to ammonia by the enzyme nitrogenase and H_2 (g) is produced (Rey et al. 2007). Ammonia produced by these reactions can then be assimilated by the organisms to produce the nitrogen-containing compounds glutamine and glutamate (White 2007). In hydrogen producing-organisms assimilation of ammonia is important since high ammonia concentrations may result in repression of nitrogenase and hydrogen production (McKinlay and Harwood 2010).

3.1. PRP-Finder algorithm

Comparative genome analysis has been successful in identifying some individual phenotype-related enzymes but has struggled to identify phenotype-related metabolic pathways. If a single enzyme is responsible for a given phenotype, then its presence in an organism will be strongly correlated to that of the phenotype (Pellegrini et al. 1999, Jim et al. 2004). However, an enzyme that is part of a metabolic pathway may be present in the organism yet the pathway is not. The presence of such enzymes may show little correlation to the presence of the phenotype.

To enable the prediction of phenotype-related metabolic pathways, we propose a phenotype-driven comparative analysis of genome-scale metabolic networks across phenotype exhibiting organisms and contrasting this comparison across organisms that do not exhibit the target phenotype. This way, we extend a comparative analysis approach to handle sets of enzymes—rather than individual enzymes—in the context of genome-scale metabolic networks. Since it is computationally intractable to consider all possible enzyme sets, some previous approaches have considered only those enzyme sets that correspond to a known metabolic pathway (Tamura and D'haeseleer 2008). However, such a methodology is limited because it cannot detect an interplay, or cross-talk, among phenotype-related metabolic pathways. This calls for approaches that could compare networks at the genome-scale rather than at the individual pathway level.

From a graph-theoretical perspective, we address a computational problem that aims to identify connected subgraphs that are both conserved and statistically enriched across metabolic networks of phenotype expressing organisms. We have developed a heuristic algorithm, called PRP-Finder that is the only known algorithm that can identify the desired subgraphs across tens and even hundreds of metabolic networks in practical time. PRP-Finder "grows" the desired subgraphs and limits the number of subgraphs grown (Schmidt and F.Samatova 2009). PRP-Finder runs in a matter of seconds, yet the subgraphs identified by the PRP-Finder are highly correlated to the subgraphs that are the most likely to represent phenotype-related metabolic pathways.

4. Conclusions

In summary, our hybrid top-down and bottom-up methodologies were able to identify phenotype-related metabolic processes and potentially important interactions among metabolic sub-networks (e.g., pathways and dense subgraphs). In both approaches, we were able to identify known enzymes and reactions associated with hydrogen production through fermentative reactions. In addition, we were able to identify potentially new interactions among the enzymes and the metabolic pathways. Overall, the information obtained from these methodologies, while preliminary, provides valuable insights about metabolic processes involved in expression of phenotypic traits important for engineering microbial organisms for enhanced bio-hydrogen production and improving upon current biotechnological applications through application of bioengineering tools.

Acknowledgements

The work of AMR was supported by the State of Florida 21st Century World Class Scholar Program, Alfred P. Sloan Foundation, and the Research Alliance in Math and Science (RAMS) program sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy. The work of MCS, WH, and NFS was supported by the by the "Ultrascale Computational Modeling of Phenotype-Specific Metabolic Processes in Microbial Communities" project from U.S. Department of Energy, Office of Science. The work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725. This work has been authored by a contractor of the U.S. Government, accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

References

Akhtar, M.K., and P.R. Jones (2008). "Engineering of a Synthetic hydF=hydE=hydG=hydA operon for biohydrogen production." *Analytical Biochemistry* **373**: 170–172.

Hendrix, W., A.M. Rocha, M. Elmore, J. Trien, and N. F. Samatova (2010). *Discovery of Enriched Biological Motifs with Knowledge Priors*. BIOCOMP'10–11th International Conference on Bioinformatics and Computational Biology.

Jones, P.R. (2008). "Improving Fermentative Biomass-derived H_2 Production by Engineering Microbial Metabolism." *International Journal of Hydrogen Energy* **33**: 5122–5130.

McKinlay, J.B., and C.S. Harwood (2010). "Photobiological production of hydrogen gas as a biofuel." *Current Opinion in Biotechnology* **21**(3): 244–251.

Nath, K., and D. Das (2004). "Improvement of hydrogen production: Various approaches." *Appl. Microbiol. Biotechnol.* **65**: 520–529.

Rey, F.E., E.K. Heiniger, and C.S. Harwood (2007). "Redirection of metabolism for biological hydrogen production." *Applied and Environmental Microbiology* **73**(5): 1665–1671.

Schmidt, M.C., and N.F.Samatova (2009). *An Algorithm for Discovery of Phenotype Related Metabolic Pathways*. IEEE International Conference on Bioinformatics & Biomedicine Washington, D.C.

Tamura, M., and P. D'haeseleer (2008). "Microbial genotype-phenotype mapping by class association rule mining." *Bioinformatics* **24**(13): 1523–1529.

Vignais, P.M., B. Billoud, and J. Meyer (2001). "Classification and phylogeny of hydrogenases." *FEMS Microbiology Reviews* **25**(4): 455–501.

White, D. (2007). *The physiology and biochemistry of prokaryotes*. New York, Oxford University Press.

Developing improved MD and QMMM codes for modeling enzymes essential to biomass liquid fuel production

M.J. Williamson,[1] A.W. Götz,[1] M.K. Garrahan,[2] E.H. Knoll,[3] C.L. Brooks III,[2] R.C. Walker,[1] and M.F. Crowley[3]

[1]San Diego Supercomputer Center, La Jolla, CA 92093, USA
[2]University of Michigan, Ann Arbor, MI 48109, USA
[3]National Renewable Energy Laboratory, Golden, CO 80401, USA

Abstract. Understanding the mechanisms and improving the efficiency of the hydrolysis of cellulose for the production of liquid biofuels is critical for establishing a sustainable energy future. The mechanism of action of cellulose-degrading enzymes is being illuminated through a multidisciplinary collaboration that uses molecular dynamics (MD) and hybrid quantum mechanics molecular mechanics (QMMM) simulations. This paper details our efforts to improve the efficiency of MD codes, as well as the addition of PM6 semiempirical hybrid QMMM functionality to such codes. Our efforts are extending the capabilities of MD codes to allow simulations of enzymes and substrates on large-scale HPC resources.

1. Introduction

Lignocellulose could potentially serve today as the dominant renewable biofuel source, capable of meeting the needs for a liquid transportation fuel, if it could be efficiently and economically depolymerized into its component glucose for microbial fermentations [1,2]. One major obstacle to the efficient and effective use of cellulose is the high resistance of crystalline cellulose to chemical and biological hydrolysis.

Given the limited experimental methods available to probe complex enzymatic cellulose depolymerization at the molecular level, we are using improved MD and QMMM simulations to explore aspects of the questions outlined above. While other presentations at SciDAC 2010 from our larger collaborative research group detail specific carbohydrate and enzyme simulation results made possible by some of the code improvements, this paper focuses on improvements in the efficiency of AMBER [3] and CHARMM [4] for such simulations, and the addition of valuable new methods for use on current and future HPC hardware.

While future efforts include developing specialized MD methods from scratch, the majority of the efforts to date involve improving two existing codes: AMBER and CHARMM. Developing a new MD code from scratch that includes the dozens of methods in the existing codes is not feasible, but improving the kernel of existing molecular dynamics engines (MDE) is tractable. Other computational codes besides AMBER and CHARMM exist for performing simulations of up to millions of atoms on as many as thousands of processors with some degree of scaling efficiency, namely, LAMMPS [5], Desmond [6], and NAMD [7]. However, these codes have some limitations and a limited number of sampling techniques. Many of the most valuable molecular modeling methods such as thermodynamic integration [8], free energy perturbation, pulling or steered MD, umbrella sampling [9], and QMMM [10,11] have been restricted to smaller molecular systems, or have not been implemented in the most scalable parallel algorithms. There is a great need for improving the scaling efficiency of MD

programs and to migrate more complex simulation and sampling methods into highly scalable programs, as well as adding additional QMMM techniques to existing fast codes. Many of the necessary methods for studying the proposed mechanisms of cellulase enzymes and the cellulose substrate are available in CHARMM.

2. CHARMM Code Modernization

Modernization of the CHARMM software from Fortran 77 to Fortran 95 has allowed the adoption of modern coding practices, which enhance the program speed, allow better compiler optimization, allow the use of modern debuggers and analysis tools, and, allow programmers to more easily modify the code for current and future parallel algorithm improvement. In August 2008, several developers began migrating the CHARMM code from Fortran 77 to Fortran 95. The effort involved 600,000 lines of code and emphasized memory allocation and common blocks. The first Fortran 95 version of CHARMM was released to the wider development community in February 2010 as c36a4 [4].

Because dynamic memory allocation was not available in Fortran 77, CHARMM originally implemented a heap and stack using large global arrays of integers. These monolithic arrays tended to waste memory, and the variables in them were declared as integer offsets requiring type conversion. Furthermore, users of 64-bit Intel machines had begun to have conflicts between pointers and integers. The standard dynamic allocation facilities of Fortran 95 allow memory to be allocated more efficiently and referenced more simply. These changes also allow such tools as Valgrind memcheck [12] to detect array boundary violations and uninitialized values.

The following modern coding practices were implemented: conversion of common blocks into modules; enclosure of some subroutines within modules, enabling compile-time checking of their argument lists; conversion of line continuations and comments from fixed-form to free-form syntax; elimination of obsolete constructs such as equivalence declarations, computed gotos, and arithmetic ifs when possible. Effort is ongoing to increase information hiding [13] by placing subroutines in the same modules as the variables they use.

In order to keep bugs from accumulating, we have employed CruiseControl [14], a continuous integration [15] tool commonly used in commercial software development. Whenever a developer commits changes to version control, CruiseControl retrieves the current sources, builds an executable, runs the regression test suite, and notifies developers of any new test failures. A script converts the test results into a format resembling that of the JUnit test framework for which CruiseControl was designed.

3. Implementation and Validation of the CHARMM Force Field in AMBER

As part of ongoing efforts to maximize the computational efficiency of current cellulose modeling, support for the CHARMM force field has been implemented in one of AMBER's MDEs, PMEMD, which has improved serial and parallel performance. Careful verification that the CHARMM force field was being implemented correctly in AMBER's MDEs was required, and a suite of tests were constructed in addition to a common file format for the comparison of energies and forces between the two MDEs.

A force field is defined by its specific potential energy equation and its specific set of associated parameters. It is independent of the MDE that it is expressed in. For a faithful reproduction of a force field that exists in a reference MDE, one needs to be able to reproduce identical total potential energy of the system and identical energy gradients for each atom. MDE attributes external to the force field (such as the thermostat, long-range electrostatic treatment, and non-bond cutoffs) must be controlled for a proper comparison.

A comparison of energy and force output between the reference CHARMM simulations to PMEMD validated the correct implementation of the CHARMM force field in AMBER. Starting with version c36a2 of CHARMM, a command (frcdump) has been implemented which outputs the various force field potential energy contributions and the energy gradient of each atom. This formatted output was compared to that generated by the AMBER MDEs. The relative error in energy (kcal/mol)

between CHARMM and PMEMD with the CHARMM force field for a test suite, as shown in Figure 1, was less than 10^{-13} kcal/mol for each of the various components of the total energy, such as bond, angle, dihedral angle, improper angle, electric, van der Waals energies. In other words, the error was near the floating point machine precision.

Figure 1. The test suite for validation included, from left to right, trialanine peptide with 33 atoms, DHFR enzyme with 2489 atoms, and a cellulose crystal fiber with 88,101 atoms solvated in TIP3P water.

4. CHARMM Performance Improvements

Profiling of performance and parallelization have been carried out on the current developmental version of CHARMM with the aim to optimize serial performance and parallel scaling. CHARMM presents the user with a range of algorithmic methods for the same calculation, which are controlled through the interplay of compile time and runtime keywords. With such diversity, we have explored the performance of the multiple available methods and the corresponding routines that are desirable for optimization efforts with the advanced profiling tools IPM [16] and TAU [17]. The testing strategy involved running a specific benchmark from the validation suite (Figure 1) over a range of builds and runtime options.

As expected, TAU profiling identified that a major part of the serial time for each thread was spent in subroutines within the EWALD module that calculates non-bonded interactions. Surprisingly, these and other routines that have been laid out in a specific form to promote the generation of faster machine code (i.e., vectorization) do not perform significantly faster than the default routines. Specifically, the REWALD95() subroutine performs as well as REWALD95_SB() and slightly better than the "Expanded fast atom list nonbond routine." This suggests that compilers are improving and the strategy of reordering code to suit the compiler's palate may be futile. Algorithmic changes are more likely to yield improvements.

The use of lookup tables results in the fastest simulations [18], but the accuracy of the energy and forces within the tables needs to be investigated. Switching the table from its default single precision to double has not improved the accuracy. Future work to improve this may involve changing the interpolation method. A quadratic or cubic spline may improve the accuracy without compromising speed.

Advanced parallel profiling with TAU confirmed that parallel scaling bottleneck in CHARMM is the particle mesh Ewald (PME) routines for calculating the non-bonded electrostatic terms at each time step. Our efforts are ongoing to redesign the core of this parallel algorithm.

5. Addition of Advanced QMMM

The development of advanced quantum mechanical (QM) methodologies for MD simulations of enzymes is essential to overcome the primary limitations associated with the use of classical molecular mechanics (MM) potential energy functions. This includes the ability to describe charge transfers and break covalent bonds and thus observe reaction events during simulations. In addition, QM methods require none or far less parameters than MM based approaches and can therefore be applied to proteins that employ co-enzymes and catalytic or structural metal centers which are typically not covered by traditional protein force fields. Because of far higher required time and lower parallel scaling efficiency of the QM Hamiltonian relative to MM, one often uses coupled QMMM potential based

simulations [10,11,19], in which the reactive or structurally important part of the protein is described with a QM based potential while the surrounding protein and solvent region is treated with an MM potential.

At present, semiempirical QM Hamiltonians based on the modified neglect of diatomic overlap (MNDO) approximation are the best compromise between numerical accuracy and computational effort for large scale biomolecular simulations. The new semiempirical hybrid QMMM implementation in AMBER [10] includes support for implicit solvent (generalized Born) simulations and for explicit solvent simulations under periodic boundary conditions via a QMMM compatible version of the PME approach. It makes use of a new automatic link atom approach for the treatment of the QMMM boundary, which does not introduce additional degrees of freedom and simplifies simulation setup. Serial and parallel improvements of the QMMM implementation in AMBER [10] have lead to a high efficiency of the code which, for an appropriately sized QM region of the order of 100 atoms, allowing computational throughput that is comparable to pure MM simulations. With AMBER QMMM support for enhanced sampling techniques specifically targeted at biomolecules, including umbrella sampling [9] with complex restraints, thermodynamic integration (TI) [8] and nudged elastic band (NEB) [20], it is possible to study rare events such as conformational changes and enzymatic reaction mechanisms for large scale systems.

The reliability and applicability of QMMM simulations have addressed with the implementation of a series of MNDO type Hamiltonians including special parameterizations for metal containing protein systems such as ZnB [21] for catalytic and structural zinc ions in protein environments. The PM6 [22] method stands out among earlier MNDO type Hamiltonians, the key difference being a modified core repulsion function with explicitly atom-pair based parameters which rectifies accuracy shortcomings and problems of transferability of atom based parameters. In addition, the PM6 parameters have been optimized for a large data set including biologically relevant molecules, which makes PM6 more suitable for biomolecular simulations than preceding methods. The PM6 Hamiltonian for all elements which do not require d orbitals has been made available with the release of version 1.4 of AMBERTools [23] and is implemented in the development version of AMBER 12 in a way that preserves the ability to run molecular dynamics in the NVE ensemble

The QMMM functionality of AMBER has been made available as a stand-alone library, which will allow easy integration into other advanced MDEs like CHARMM. Work is under way for the inclusion of d orbital support for PM6 and other advanced MNDO type Hamiltonians to extend the applicability of the QMMM functionality to proteins containing hypervalent elements and transition metals. We are also planning to couple the ADF program [24], which shows good parallel scaling due to extensive use of numerical quadrature schemes, with the AMBER QMMM module to make it possible to use density functional theory (DFT) based QM potentials in MD simulations which provide better accuracy and reliability and a more general applicability than semiempirical methods. Problems of the applicability of QMMM MD simulations when solvent molecules enter or leave the QM region are addressed with ongoing work on the implementation of adaptive QMMM methods, which allow variable QM regions during QMMM MD simulations.

Acknowledgments

This work is performed with support from the U.S. Department of Energy Office of Science through a SciDAC award titled “Understanding the Processivity of Cellobiohydrolase I.” Partial support for the work described here also was contributed by the NIH through a grant to CB (RR023920).

References

[1] Himmel, M. E. (2008). Biomass recalcitrance : deconstructing the plant cell wall for bioenergy. Oxford: Blackwell Pub.

[2] Lynd, L. R., Laser, M. S., Brandsby, D., Dale, B. E., Davison, B., Hamilton, R., et al. (2008). How biotech can transform biofuels. Nature Biotechnology, 26(2), 169–172.

[3] Case, D. A., Cheatham, T. E., Darden, T., Gohlke, H., Luo, R., Merz, K. M., et al. (2005). The Amber biomolecular simulation programs. Journal of Computational Chemistry, 26(16), 1668–1688.
[4] Brooks, B. R., Brooks, C. L., Mackerell, A. D., Nilsson, L., Petrella, R. J., Roux, B., et al. (2009). CHARMM: The Biomolecular Simulation Program. Journal of Computational Chemistry, 30(10), 1545–1614.
[5] Plimpton, S. (1995). Fast Parallel Algorithms for Short-Range Molecular-Dynamics. Journal of Computational Physics, 117(1), 1–19.
[6] Bowers, K. J., Chow, E., Xu, H., Dror, R. O., Eastwood, M. P., Gregersen, B. A., et al. (2006). Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters. Paper presented at the ACM/IEEE Conference on Supercomputing (SC06).
[7] Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., et al. (2005). Scalable molecular dynamics with NAMD. Journal of Computational Chemistry, 26(16), 1781–1802.
[8] Simonson, T. (2001). Computational biochemistry and biophysics. New York: Marcel Dekker.
[9] Torrie, G. M., and Valleau, J. P. (1977). Non-Physical Sampling Distributions in Monte-Carlo Free-Energy Estimation - Umbrella Sampling. Journal of Computational Physics, 23(2), 187–199.
[10] Walker, R. C., Crowley, M. F., and Case, D. A. (2008). The implementation of a fast and accurate QM/MM potential method in Amber. Journal of Computational Chemistry, 29(7), 1019–1031.
[11] Warshel, A., and Levitt, M. (1976). Theoretical Studies of Enzymic Reactions—Dielectric, Electrostatic and Steric Stabilization of Carbonium-Ion in Reaction of Lysozyme. Journal of Molecular Biology, 103(2), 227–249.
[12] Valgrind Developers (2009). Valgrind (Version 3.5) [Software]. Available from http://valgrind.org/.
[13] McConnell, S. (2004). Code Complete (2nd ed.). Redmond, Wash.: Microsoft Press.
[14] ThoughtWorks Inc. (2009). CruiseControl, a Continuous Integration Toolkit (Version 2.8) [Software]. Available from http://cruisecontrol.sourceforge.net/.
[15] Fowler, M. (2006). Continuous Integration, from http://www.martinfowler.com/articles/continuousIntegration.html.
[16] Write, N. J., Pfieffer, W., and Snavely, A. (March 2009). Characterizing Parallel Scaling of Scientific Applications using IPM. Paper presented at the The 10th LCI International Conference on High-Performance Clustered Computing. from http://ipm-hpc.sourceforge.net/.
[17] Shende, S. S., and Malony, A. D. (2006). The TAU parallel performance system. International Journal of High Performance Computing Applications, 20(2), 287–311.
[18] Nilsson, L. (2009). Efficient Table Lookup Without Inverse Square Roots for Calculation of Pair Wise Atomic Interactions in Classical Simulations. Journal of Computational Chemistry, 30(9), 1490–1498.
[19] Senn, H. M., and Thiel, W. (2009). QM/MM Methods for Biomolecular Systems. Angewandte Chemie-International Edition, 48(7), 1198–1229.
[20] Mills, G., Jonsson, H., and Schenter, G. K. (1995). Reversible Work Transition-State Theory - Application to Dissociative Adsorption of Hydrogen. Surface Science, 324(2–3), 305–337.
[21] Brothers, E. N., Suarez, D., Deerfield, D. W., and Merz, K. M. (2004). PM3-compatible zinc parameters optimized for metalloenzyme active sites. Journal of Computational Chemistry, 25(14), 1677–1692.
[22] Stewart, J. J. P. (2007). Optimization of parameters for semiempirical methods V: Modification of NDDO approximations and application to 70 elements. Journal of Molecular Modeling, 13(12), 1173–1213.

[23] AMBER (Version 11) (2010). [Software]. Available from http://www.ambermd.org.
[24] ADF, SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands, Available from http://www.scm.com.

High-performance computer simulation in the energy biosciences

Jiancong Xu,[1,2] Mike F Crowley,[2,3] Loukas Petridis,[1,2] Hao-Bo Guo,[1] Hong Guo,[1,4] Marimuthu Krishnan,[1] Benjamin Lindner,[1,4] Anne O Summers,[5] Susan M Miller,[6] Liyuan Liang,[7] Jerry M Parks,[1] Roland Schulz,[1,4] Moumita Saharay,[1,2] Xiaolin Cheng[1,2] and Jeremy C Smith[2,4,*]

[1]Center for Molecular Biophysics, Building 6011, Oak Ridge National Laboratory, Oak Ridge, TN, 37830, USA
[2]BioEnergy Science Center
[3]Chemical and Biosciences Center, National Renewable Energy Laboratory, 1617 Cole Blvd, Golden, CO, 80401-3393, USA
[4]Department of Biochemistry & Cellular & Molecular Biology, University of Tennessee, Knoxville, TN 37966, USA
[5]Department of Microbiology, University of Georgia, Athens, Georgia, 30602-2605
[6]Department of Pharmaceutical Chemistry, University of California San Francisco, 600 16th Street, San Francisco, California, 94158
[7]Environmental Science Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 37831

Email: *smithjc@ornl.gov

Abstract. An overview is provided of the use of high-performance computing for molecular simulations in the energy biosciences. Examples are given of research in cellulosic ethanol production, involving the construction of models of lignocellulosic biomass and the characterization of hydrolytic enzyme complexes. The use of quantum mechanical methods in determining enzyme reaction mechanisms is discussed. The synergy between neutron scattering and computer simulation is examined. Finally, MD scaling results on the ORNL Jaguar Cray XT5 supercomputer are given.

1. Introduction

High-performance computer simulation has a significant role to play in the energy biosciences in obtaining an understanding of physical processes leading to biological function. Molecular mechanical and quantum mechanical techniques can provide atomic detailed insight into processes at the core of research into bioenergy, bioremediation, carbon capture, neutron scattering and other critical research missions.

Here we will provide a synopsis of some current projects at Oak Ridge. Work is in progress to derive computational simulation models of use in cellulosic ethanol production and models are being constructed of heterogeneous lignocellulosic biomass, an effort that will hopefully shed light on the phenomenon of 'biomass recalcitrance,' or resistance, to hydrolysis into sugars, which is a bottleneck in biofuel production. Further biofuel research involves simulation work on the functioning of microbial enzymes and enzyme complexes that hydrolyse cellulose chains. The fate of mercury in

streams contaminated from DOE sites is strongly influenced by bacterial enzymes, and we outline research aimed at understanding how these enzymes function. Neutron scattering will play a significant role in the energy-related materials and biosciences, and the synergy between neutron scattering and HPC is described. Finally, we present some scaling results for MD of large systems on a petascale supercomputer.

2. Bioenergy Research

2.1. Lignocellulosic Biomass

This section outlines the construction of an all-atom computational model of lignocellulose, a complex material found in plant cell walls. Lignocellulosic biomass is composed of crystalline cellulose microfibrils laminated with hemicellulose, pectin, and lignin polymers [1].

Initial work has focused on cellulose and lignin. Cellulose is the unbranched β-(1–4)-linked polymer of D-glucopyranose [2]. It exists as an array of many parallel, oriented chains, organized into crystalline microfibrils [3]. The glucan chain length varies from about 2000 to more than 15,000 glucose residues. The organization of cellulose can vary from elementary fibrils in plants, which contain approximately 36 chains, to large macrofibrils of cellulosic algae, which contain more than 1200 chains. Lignins are heterogeneous branched biopolymers [4]. The chemical composition and structure of lignins are highly heterogeneous, varying significantly between different plant species and even within different parts of the same plant wall. Unlike the other components of the plant cell wall, lignins do not have a well-defined primary structure.

In recent years there has been a revived interest in lignocellulose structure, due to its immense abundance in nature offering a potentially cheap sugar source for industrial bioethanol production [5]. However, the conversion efficiency of lignocellulose into ethanol in integrated enzyme-based industrial processes is low. The bottleneck in the process is the hydrolysis of biomass into sugars: the intermolecular interactions between the lignocellulosic components may prevent efficient decomposition [5–7]. Therefore, understanding the physical properties of this biomass is crucial for overcoming the major technological challenge in the development of viable cellulosic bioethanol. Computer simulation is a powerful tool for complementing experiment in obtaining an understanding of the molecular-level structure and dynamics of lignocellulose. Molecular Dynamics (MD) simulation may provide valuable insights on the molecular-level structure, dynamics and energetics of this complex biomass system.

2.1.1. A molecular mechanics force field for lignin. In recent work [8] we presented the first essential step towards the accurate computer simulation of lignin: the derivation of an empirical molecular mechanics (MM) force field. Together with the existing force field for polysaccharides [9], this force field will enable full simulations of lignocellulose.

We outline the general strategy employed to obtain the lignin force field. The CHARMM potential energy function [10] of a molecule is given by the following equation:

$$E=\sum_{bonds}K_b(b-b_o)^2+\sum_{angles}K_\theta(\theta-\theta_o)^2+\sum_{U-B}K_{ub}(s-s_o)^2+\sum_{dihedrals}K_\phi[1+\cos(n\phi-\delta)]+\sum_{impropers}K_\psi(\psi-\psi_o)^2+$$
$$\sum_{non\text{-}bonded}\left\{\varepsilon_{ij}\left[\left(\frac{R_{ij}}{r_{ij}}\right)^{12}-\left(\frac{R_{ij}}{r_{ij}}\right)^{6}\right]+\frac{q_iq_j}{4\pi r_{ij}}\right\},$$

where the contributions to the energy include bonded (bond, angle, Urey-Bradley, dihedral and improper dihedrals) and non-bonded (Lennard-Jones 6–12 potential for the van der Waals interactions, and Coulomb electrostatic) terms. The parameters, including the force constants K and partial atomic charges q, are molecule-dependent and must be derived prior to performing simulations. The parameterization of the force field for lignin was the main task of Ref. [8].

This parameterization of lignin followed the main procedure of parameterization of proteins [11]. Two model compounds were used to save computational time (see Figure 1): methoxybenzene, and p-hydroxyphenyl (PHP), the simplest lignin unit. Parameters were optimized by considering two factors. Firstly, the 'target data' were reproduced as closely as possible. Effectively this ensures that the force field describes accurately specific properties of lignin. Secondly, compatibility with the existing CHARMM force field was ensured by restricting optimization to parameters that did not already exist in the force field.

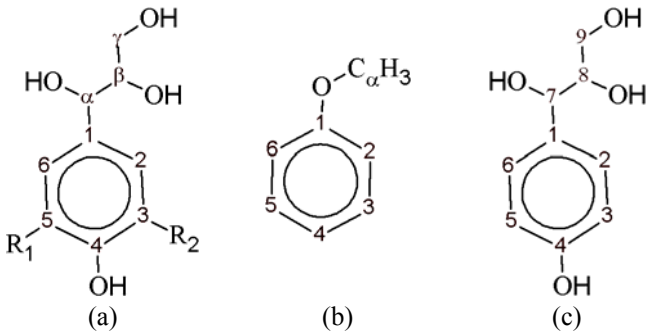


Figure 1. (a) Three lignin units: p-hydroxyphenyl, where R_1 = H and R_2 = H. Guaiacyl, where R_1 = H and R_2 = OMe. Syringyl, where R_1 = OMe and R_2 = OMe. (b) methoxybenzene and (c) p-hydroxyphenyl (PHP).

The optimization strategy for the new parameters is summarized in the diagram in**Error! Reference source not found.**. Equilibrium values for bond lengths, angles and dihedrals were taken from high level (MP2/6-31G*) QM-optimized geometries and were not further revised. The van der Waals parameters were taken unaltered from the existing protein [11] and ether [12] CHARMM force fields. Charges were optimized with respect to the QM interaction energies using a supramolecular approach of methoxybenzene interacting with one water molecule. The partial charges were adjusted so as to reproduce minimum distances and interaction energies between anisole and water. In order to mimic the effect of electronic polarizability, which is not explicitly taken into account in additive force fields, atomic charges were purposely overestimated. The empirical calculations were found to reproduce well the scaled QM interaction energies, with the error being less than 3%. After completing the non-bonded interactions, parameters for dihedral rotations were deduced from QM potential energy surfaces. The optimization was based on reproducing quantum-chemically obtained adiabatic energy surfaces, where the selected dihedral is held constant while the remaining degrees of freedom are allowed to relax to a constrained energy minimum. The remaining bonded parameters (bonds and angles) were optimized to reproduce vibrational frequencies and eigenvector projections derived from QM calculations. For this the Automated Frequency Matching Method [13] was employed, which optimizes the MM parameter set until the best fit with the QM reference set was obtained. Finally, the parameter set was tested without further adjustment against condensed phase experimental properties of lignin that were not used during the parameterization.

2.1.2. Lignocellulose models. The next step towards a realistic simulation of lignocellulosic biomass is the construction of an atomistic model. This is performed by first generating models of individual components, such as lignin and cellulose (hemicellulose and pectins will be incorporated in the model in the future) and then combining these components to form supramolecular assemblies.

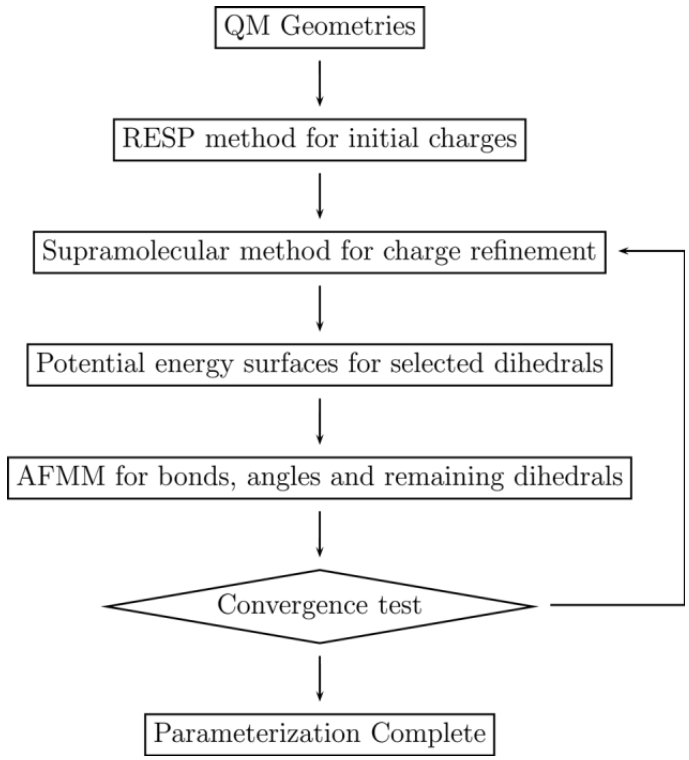


Figure 2. Schematic representation of the strategy employed to parameterize the force field for lignin.

The accurate computer simulation of lignin presents significant challenges. Unlike many biological macromolecules that have been studied with molecular simulation, both the primary and three-dimensional structures of lignins are not known. However, there is abundance of information on lignin composition. Lignins are composed primarily of three units p-hydroxyphenyl (H), guaiacyl (G), and syringyl (S). Softwoods, in contrast, contain mainly only G units [4]. There are various linkages that connect the units, leading to the formation of the branched lignin biopolymer. The typical linkage composition of softwoods is (Pu08): β-O–4’ 50%, 5–5' 30%, α-O–4' 10% and β–5' 10%. Crosslinks are formed when one unit participates in more than one linkage. The crosslink density of spruce wood lignins has been deduced via calculations on extractive delignification [14] and was found to be 0.052. Finally, lignins are known to be optically inactive [15].

The lignin models were constructed by first deriving the topology of the molecules and then generating their 3D structures (Figure 3). With the above experimental data as a guide, random configurations of lignins were created producing 26 molecules with different topologies, but all consistent with experimentally-determined properties of softwood. For example, although all lignins had the same linkage composition, the order of the linkages was different. Furthermore, the number of crosslinks varied between zero and six and their positions along the lignin molecule were also different. Once topologies were derived, the 3D structures for the lignin molecules were constructed. A step-wise approach was used, where each new unit is first added to the existing structure using the appropriate linkage and subsequently the entire new molecule was optimized using the molecular mechanics function.

In contrast to lignins, structures of cellulose microfibrils [16] or and Iβ qre known [3]. In the present model, as in other studies [2,17], cellulose is in the Iβ form and the MD simulation starts from a configuration in which the fibril is a perfect crystal. The final step in the generation of an initial model of lignocellulose is to surround the cellulose fibril with the lignin molecules.

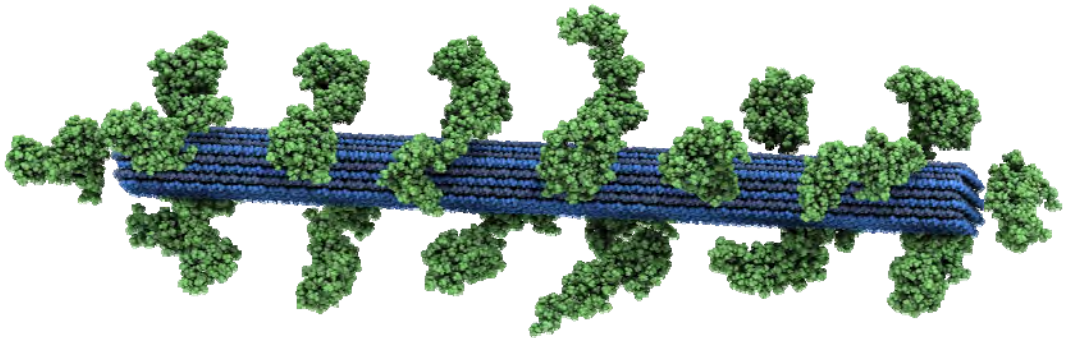


Figure 3. Atomistic model of lignocellulose of softwoods. The cellulose elementary microfibril is shown in blue and the 26 lignin molecules are shown in green. The total system size in 3.3 million atoms (water molecules not shown for image clarity).

2.2. Towards Structure-Based Cellulosome Design for Cellulosic Ethanol

One promising avenue for overcoming biomass recalcitrance is to understand and modify the properties of bacterial cellulosomes [18]. The cellulosome is a large extracellular multienzyme complexes produced by anaerobic bacteria for the efficient degradation of plant cell-wall polysaccharides [19]. The cellulosome complex consists of various kinds of enzymes arranged around a non-catalytic scaffolding protein that enables the complex to adhere to the biomass and bacterial cell surface. The most extensively studied cellulosome system belongs to *Clostridium thermocellum,* in which the incorporation of the cellulosomal enzymes into the complex is accomplished by the high affinity interaction between two complementary modules: the type I cohesin modules within the scaffolding protein and the type I dockerin domain carried by cellulosomal enzymes.

The crystal structure of the type I cohesin-dockerin complex from *C. thermocellum* has been determined [20] (Figure 4). The cohesin domain exhibits an elongated nine-stranded β-barrel in a classical jelly-roll topology, while the dockerin partner is organized into two symmetrically oriented calcium-binding loop-helix motifs. The cohesin-dockerin association is maintained mainly by hydrophobic interactions, augmented through an extensive hydrogen-bonding network between one face of the cohesin and the corresponding dockerin domain. A number of hydrophilic residues play an essential role in the recognition and formation of the complex: Arg77, Tyr74, Asp39, Glu86 and Gly89 of the cohesin domain, and Leu22, Arg23, Ser45, Thr46 and Arg53 from α-helices 1 and 3 of the dockerin domain (Figure 4).

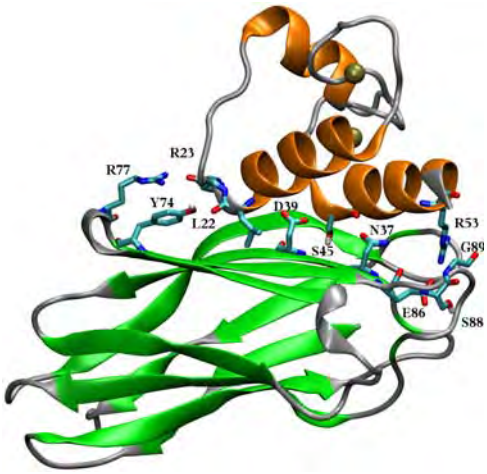


Figure 4. Crystal structure of the cohesin-dockerin complex in cartoon representation with β-sheets (cohesin) in green, α-helices (dockerin) in orange and loop regions in silver.

2.2.1. Atomistic molecular dynamics simulations of cohesin-dockerin complexation. Recognition of Type I cohesins by dockerins is the determining event in assembly of individual enzymatic subunits into the cellulosome complex. Although the crystallographic structure and experimental measurements have provided essential information about the association of cohesins and dockerins, the underlying microscopic dynamic and energetic processes are not directly

accessible to experiments. It is therefore particularly informative to elucidate computationally the detailed molecular principles upon which the cohesin-dockerin interaction is based at the atomic level.

The understanding of the underlying molecular association/dissociation mechanism in terms of structural and dynamical events is facilitated by the knowledge of the free energy profile for the WT cohesin-dockerin dissociation. The free energy of cohesin-dockerin dissociation was estimated from a total of 100 ns MD simulation in bulk solution, using the adaptive biasing force (ABF) method [21,22] implemented in NAMD [23]. This method relies upon the integration of the average force acting on the reaction coordinate (ξ) obtained from unconstrained MD simulations. The free energy profile was then obtained by allowing the two domains to diffuse reversibly along the relative center-of-mass reaction coordinate. The results are shown in **Error! Reference source not found.**.

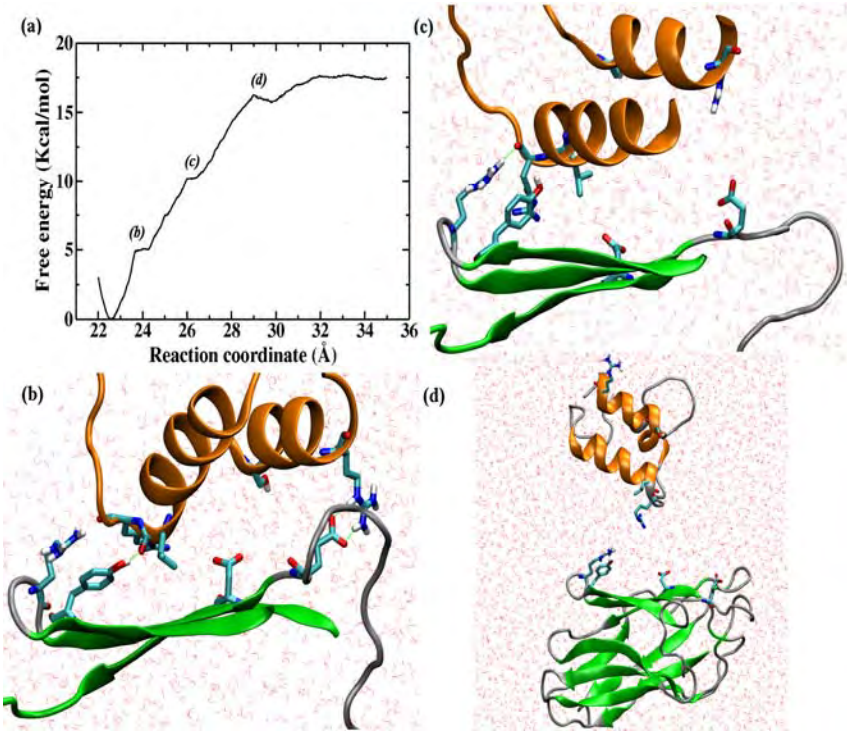


Figure 5. (a) Free energy profile for the dissociation of cohesin and dockerin domains, (b) ξ = 24 Å, and (c) ξ = 27 Å; (d) ξ > 30 Å.

The overall shape of the free energy profile along the reaction coordinate exhibits a general uphill trend, illustrating quantitatively that the cohesin-dockerin complex exhibits a resistance against external forces and that there is a high affinity for the two domains to remain bound. As the two domains move away from the stable bound state (22.5 Å), the cohesin-dockerin interactions are progressively disrupted. Firstly, this leads to a steep increase of the free energy before reaching the first shoulder at ~24 Å, at which point residues Asp39 and Ser45 at the interface of the protein complex are no longer in contact (Figure 5b) and water flows into the binding area to fill up the space. As the two domains move further apart, the free energy profile reaches the second slight shoulder at ~26 Å. Inspection of the simulation trajectory indicated that the second shoulder corresponds to the disruption of the recognition strip interaction with the C-terminal region of α-helix 3, accompanied by the rupture of hydrogen bonds/salt bridges between Arg53 and Glu86 (Figure 5c). In contrast, at this point of the dissociation the C-terminal of the first α-helix of the dockerin is still repeatedly in contact with the side chains of the solvent-exposed residues in the β–strand 5/6 loop at the other end of the β-barrel. The ultimate dissociation of the interactions corresponds to the shallow well emerging at ~30 Å before the PMF eventually becomes nearly flat at >35 Å (Figure 5d).

The sequential events of interdomain hydrogen-bond rupture and the step-by-step pattern of the cohesin-dockerin dissociation revealed by the free energy calculations provide evidence for a mode of binding involving both α-helices in the dockerin and the corresponding surface region from its cohesin partner: the C-terminal end of dockerin helix 1 interacting with the β-strand 5/6 loop, while the N-terminus diverted away from the cohesin surface; the N-terminus of helix 2 covering the core fragment of the β-sheet interface and the C-terminus interacting with the recognition strips. The cohesin-dockerin binding may therefore take place in a cooperative manner. Although examination of the crystal structure alone suggests that the formation of the cohesin-dockerin complex involves relatively large surface areas on both partners, the present results show that specific surface regions play more critical roles than others in forming and maintaining the integrity of the cellulosome complex.

2.3. Cellulases

Cellulases are a family of enzymes that catalyze the hydrolysis of celluloses, the main component of the plant cell wall, thus converting the polysaccharides to sugars that can be further fermented to renewable energy sources such as ethanol and other chemicals [24–26]. Three important types of enzymes are included in this family, endoglucanases that cleave glycosidic bonds of cellulose by producing chain ends, exoglucanases that break a single fiber into smaller sugars, and β-glucosidases that hydrolyse simple sugars [27]. Depending on their catalytic mechanisms, cellulases can be further classified into two categories, i.e., the inverting enzymes that invert the chirality of the anomeric carbon, and the retaining enzymes that retain the anomeric configuration [28–30]. A number of cellulase crystals or solution structures are available in the Protein Data Bank [31]. The detailed catalytic mechanisms of cellulases, however, have yet to be elucidated. Computational methods are of importance in meeting the challenge of understanding the mechanistic details of the biodegradation of crystalline cellulose and plant biomass.

Quantum mechanical/molecular mechanical (QM/MM) simulation methods [32] are an effective way for to characterize enzyme reactions. QM/MM can also be applied to study the dynamic properties of macromolecules and energetics along the reaction pathways. In QM/MM only the reaction center is considered quantum mechanically, the rest being trated with MM.

Error! Reference source not found. shows the Michaelis complex structure of a retaining cellulase *Humicola grisea* Cel12A with a cellulose tetramer (cellotetraose) bound to the –2 to +2 subsites (pdb code 1W2U, resolution 1.52 Å) [33]. The glucose unit at subsite –1 is believed to undergo conformational change upon binding to the enzyme and during the catalytic reaction [34,35]. The two steps in the retaining reaction mechanism involve glycosylation and deglycosylation. QM/MM MD and free energy simulations can provide information on both the conformational changes of the glucose rings along the reaction pathway and the reaction steps, together with the free energy as a function of the reaction coordinate. Moreover, the roles of active site residues can be explored through mutagenesis studies.

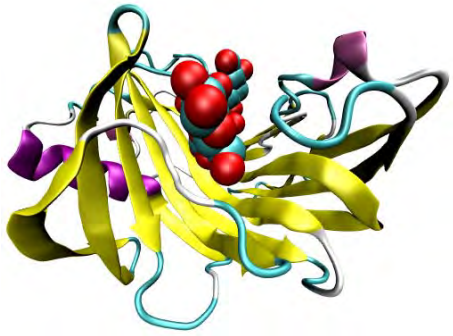


Figure 6. Structure of H. grisea Cel12A (ribbons) in complex with a cellotetraose (spheres), occupying the –2 to +2 sites at the substrate binding cleft, pdb code 1W2U.

3. Bioremediation: Mechanisms of Bacterial Mercury Resistance

Mercury is a toxic and highly reactive element that has no known biological function. It is readily interconverted between its elemental Hg(0) form and more toxic Hg(II) species through various microbial and abiotic pathways [36]. Mercury-resistant bacteria are able to transform inorganic and organic Hg(II) to less toxic Hg(0) via proteins and enzymes of the *mer* operon. The narrow-spectrum *mer* operon includes a regulatory protein, MerR, a periplasmic transporter, MerP, one or more

transmembrane Hg(II) transporters (MerT, MerC), and a mercuric reductase, MerA. Additionally, broad-spectrum mercury resistant bacteria include an organomercurial lyase, MerB.

3.1. Conformational Changes Upon Hg(II)-Binding in MerR

MerR, the first protein identified in a family of transcriptional activators known as the MerR family, is the key regulator of the *mer* operon [37,38]. MerR activates transcription of other *mer* proteins in the presence of Hg(II) in nanomolar concentrations, and represses transcription in its absence. The functional form of MerR is a homodimer that consists of a DNA binding domain, a helix-turn-helix motif, and a long, antiparallel coiled-coil. Although MerR contains two binding sites, only a single Hg(II) cation is bound at a time [38]. MerR is known to bind Hg(II) in a trigonal planar geometry in which three Cys residues coordinate to Hg(II) [39]. No conventional NMR structures have been determined for MerR thus far, due to its tendency to aggregate [40]. Although other MerR family proteins have been crystallized [41], no X-ray structures have been solved yet for MerR. At present, the detailed mechanism of allosteric transcriptional activation upon Hg(II) binding is not well understood.

Initially, MerR is bound to its operator DNA in complex with RNA polymerase (RNAP) such that the DNA is bent away from and inaccessible to RNAP [42]. Binding of Hg(II) induces an allosteric conformational change that enables transcription of all *mer* proteins. In an effort to determine the conformational changes of MerR upon Hg(II) binding, we are performing MD simulations on homology-modelled structures of Hg(II)-bound and Hg(II)-free MerR. The dynamic structural models are validated by comparison with small-angle X-ray scattering (SAXS) data in solution, which provides the molecular envelope shape, the pair distribution function, and the radius of gyration.

3.2. Enzymatic Mechanism of Hg-C Cleavage Catalyzed by the Organomercurial Lyase MerB

Methylmercury, $[CH_3Hg(II)]^+$, is a particularly toxic organomercurial species that bioaccumulates in living organisms [43]. Its extremely high affinity for thiols and its ability to cross the blood-brain barrier are major contributing factors to its toxicity. As a result, methylmercury degradation is an important topic in bioremediation. The organomercurial lyase, MerB, catalyzes the cleavage of Hg-C bonds in organomercurials to yield Hg(II) and a hydrocarbon (Scheme 1). A separate enzyme, the NADPH-dependent flavoenzyme, MerA, then catalyzes the reduction of Hg(II) to Hg(0).

$$[\text{R-Hg(II)}]^+ \xrightarrow{\text{MerB}} [\text{Hg(II)}]^{2+} + \text{R-H}$$

Scheme 1. Reaction catalyzed by MerB (R = alkyl, aryl).

MerB functions as a monomer, and no cofactors or other prosthetic groups are involved in the reaction, but free thiols such as cysteine or glutathione are required for enzymatic turnover [44]. The active site of MerB contains two essential Cys residues [45] and an Asp residue [46]. Various mechanisms for MerB have been proposed in the literature [45–50], but there is still debate over the precise mechanism involved [51].

In our laboratory, efforts are underway to determine the detailed reaction mechanism of Hg-C cleavage in MerB. Active site models of MerB have been constructed from available X-ray structures [46]. Density functional theory (DFT) methods [52] are used to distinguish between likely and unlikely reaction pathways and to determine the specific chemistry involved in achieving catalysis by MerB.

4. Synergy of Neutron Scattering and High-Performance Computing.

The characterization of the structure and internal dynamics of biomolecules such as proteins and biopolymers is essential to understanding the mechanisms of their biological functions [53]. A combination of molecular dynamics simulations and neutron scattering techniques has emerged as a promising synergistic pathway to elucidating atomistic details of the dynamics of various biological

systems. We review here one aspect of neutron scattering in biology: recent applications in characterizing protein physics, and, in particular, the protein glass transition and protein hydration dynamics [54–70].

In neutron scattering experiments, neutron beams of suitable wavelengths and energies are directed at a sample. Neutrons are scattered by atomic nuclei of the sample. The resulting change in the energy and momentum of incident neutrons provides information about the structure and internal dynamics of the molecules concerned [71]. In MD simulation, the time evolution of the positions and momenta of atoms of biomolecules are tracked by solving the Newtonian equation of motion, taking into account the anharmonic interactions between atoms. The space and time correlation functions calculated from MD trajectories can then be used to interpret experimental neutron data [53,71].

In neutron scattering experiments, the number of neutrons scattered within a solid angle between Ω and $\Omega+d\Omega$ with a change in energy $\hbar\omega$ and momentum $\hbar\mathbf{Q}$ is measured. This number is proportional to the double-differential cross-section $\delta^2\sigma/\delta\Omega\delta\omega$, which in turn is proportional to the dynamic structure factor, $S(\mathbf{Q},\omega)$, as shown below,

$$\frac{\partial^2\sigma}{\partial\Omega\partial\omega} \;\alpha\; S(Q,\omega)$$

The dynamic structure factor can be written in terms of van Hove function, $G(\mathbf{r},t)$, which characterizes the space-time correlation of individual atoms as well as between the motions of pair of atoms, as shown in Figure 7 [71]. It is evident from Figure 7 that Fourier transformation of physical quantities that quantifies the nature of atomic dynamics in the "Molecular Dynamics Space" leads to information determined in "Neutron scattering space" and vice versa.

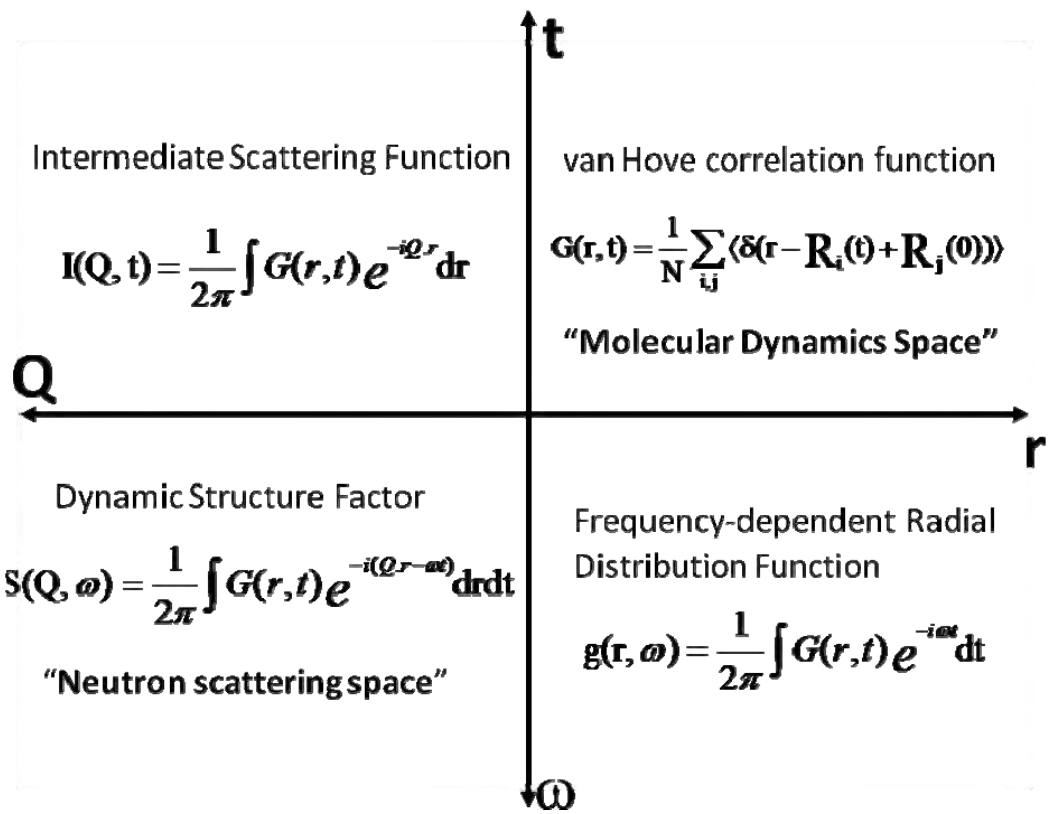


Figure 7. 4D representation of the relationship between neutron scattering functions (determined from experiments) and space-time correlation functions (calculated using MD simulations) is shown here. Here, $R_i(t)$ denotes the atomic coordinates of ith atom at time t.

A number of neutron scattering and other experimental and computer simulation studies have revealed that proteins exhibit a temperature-dependent dynamical transition around 180–220 K [54–70]. Below this temperature, the dynamics is similar to that of a glassy material, while at temperatures above 220 K, protein atoms exhibit liquid-like dynamics. Many proteins exhibit this behavior, and correlations have been observed between the onset of protein function, such as ligand binding, proton pumping or electron transfer, and the onset of the anharmonic dynamics at around 220 K. In some cases proteins may function only when the temperature is above the dynamical transition temperature. The transition has also been shown to be sensitive to changes in solvent conditions. For example, proteins immersed in viscous solvents, such as trehalose, exhibit no transition.

The experimental observations led to some important questions concerning the microscopic dynamical details of the protein glass transition, including whether the dynamical transition in a solvated protein is controlled by the solvent and whether proteins exhibit intrinsic anharmonic dynamics much below the glass transition temperature.

Neutron scattering experiments and molecular dynamics simulations have shown signatures of anharmonic dynamics below the ~220 K dynamical transitions for several proteins. The dynamic processes associated with this low-temperature anharmonicity, and how these motions may be related to global dynamical changes at the dynamical transition, have yet to be fully understood.

The mean square displacement (MSD) of atoms of hydrated myoglobin as a function of temperature calculated from MD simulation is shown in Figure 8. The MSD increases linearly at low temperatures then exhibits two slope changes: one at ~150 K and the other at ~220 K. The significant change in gradient at ~220 K is the solvent-driven dynamical transition as observed in many biological systems. At 150 K, the rotational excitations of methyl groups are observed, and these jump-like motions of methyl protons will lead to quasi-elastic neutron scattering of the type that has been observed at around 150 K in proteins. In lysozyme, the low-temperature anharmonicity was observed at 100 K and was attributed to the onset of methyl dynamics [67,68]. It was also demonstrated that the

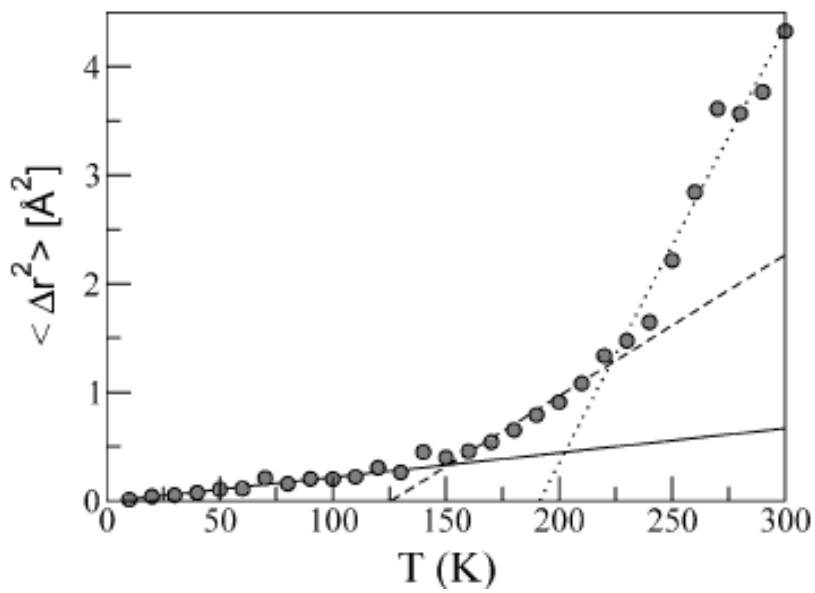


Figure 8. Time-averaged mean square displacement of myoglobin as a function of temperature. The straight lines are fits to the data for different temperature ranges (solid line, 0 to 150 K; dashed line, 150 to 220 K; dotted line, above 220 K) and are shown as a guide to the eye. (Reproduced with permission from J. Phys. Chem. B 2008, 112, 5522–5533. Copyright 2008 Am. Chem. Soc.)

anharmonic dynamics observed at ~100 K is independent of hydration level, while the dynamical transition at ~200–220 K is observed only at hydration levels greater than 0.2 g of water/g of protein. In incoherent neutron scattering experiments the main contribution to the scattered protein intensity arises from the nonexchangeable hydrogen atoms, and a significant fraction of nonexchangeable hydrogens in proteins reside on CH_3 groups: 26% in lysozyme, for example. Thus, the CH_3 groups would be expected to contribute significantly to the scattered intensity. It has been suggested that a dominant contribution of the relaxation observed in dry myoglobin neutron scattering is due to methyl dynamics. ^{1}H NMR relaxation studies have also investigated the reorientational dynamics of C-H bond vectors of methyl groups, and 1H NMR experiments on dry lysozyme have shown that 70% of the total proton relaxation is due to methyl dynamics.

In conclusion, simulation models are tested by calculating neutron scattering structure factors and comparing the results directly with experiments. If the scattering profiles agree the simulations are used to provide a detailed decomposition and interpretation of the experiments, and if not, the models are rationally adjusted. Comparison with neutron experiment is at the level of the scattering functions and also of quantities derived from them. The end result is simulation models consistent with the neutron data that can be used to design a wide range of experiments. In the above work, neutron scattering experiments and MD simulations have demonstrated the non-negligible role of intrinsic anharmonicity of protein dynamics in protein glass transition behavior.

5. Extreme Scale Supercomputers

Current supercomputers, such as the Jaguar Cray XT5 at Oak Ridge National Laboratory, are beginning to assemble over 10^5 cores and reach petaflop nominal speeds. A challenge for MD is to achieve efficient scaling up to $O(10^4–10^5)$ cores. The simulations are limited by the parallel efficiency of the MD algorithms i.e., their ability to run in parallel in many thousand processors.

The computationally most demanding part of an MD simulation of biological systems is the treatment of the long range Coulomb interactions. A common method for this is to directly calculate the Coulomb interaction for any pair of atoms separated by less than another cut-off distance, R_{coul}, and outside this distance to calculate the electrostatic interactions with the Particle Mesh Ewald (PME) method [72,73]. However, full electrostatic treatment with the PME method limits the performance on massively parallel computers.

An alternative method, that avoids the electrostatics bottleneck, is the use of the Reaction Field (RF) [74]. In this method it is assumed that any given atom is surrounded by a sphere of radius, R_{rf} within which the electrostatic interactions are calculated explicitly. Outside the sphere the system is treated as a dielectric continuum. Scaling benchmarks on a 3.3 million atom lignocellulosic biomass system show that the use of the RF drastically improves the parallel efficiency of the algorithm relative to PME, yielding 28ns/day (Figure 9) [75].

6. Conclusions

The work in progress involves both methodological development and applications to systems of prime interest in the energy biosciences. Simulations of the lignocellulosic systems described above are currently being performed on local clusters and DOE supercomputers, the latter with the help of a DOE INCITE allocation.

The enzyme simulation results have provided insight into the molecular principles that govern cohesin-dockerin domain recognition in cellulosome assembly. These principles could in turn be used to guide protein engineering modifications so as to alter cohesin-dockerin binding. Efforts are underway to design engineered cellulosomal modules which can conduct more efficient biomass degradation than the corresponding wild-type protein complexes. Both atomic-detail and coarse-grained computer simulations are expected to provide a foundation for understanding the principles of domain synergy and cellulosomal activity, thus allowing the rational, structure-based design of improved cellulosomal assemblies, a key step to attaining more cost-effective cellulosic ethanol.

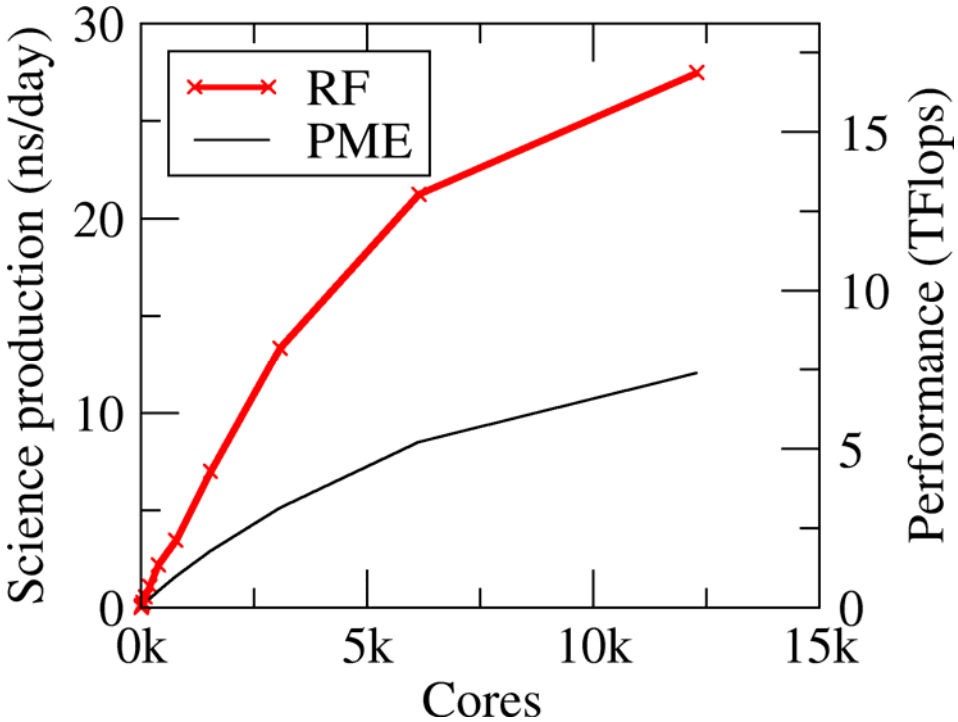


Figure 9. Strong scaling (i.e., increasing the number of cores with fixed system size) of the 3.3 million atom lignocellulose model shown in Figure 3.

With the QM/MM methodology and the high-performance computer simulations outlined above, an understanding of cellulase catalysis is being obtained that may be of practical use in designing new, efficient enzymes for industrial purposes. Similarly, an understanding of the catalytic mechanisms behind the activity of proteins in the mer operon could potentially lead to the synthesis of small-molecule mimics of use in remediation of mercury-contaminated sites.

Finally, the excitement generated by the coming on line of petascale supercomputers and next-generation neutron sources, such as the Jaguar Cray XT5 and the Spallation Neutron Source, both at ORNL, arises largely from the capabilities these facilities provide in extending the time- and length-scales accessible to detailed physical characterization. We look forward to participating in future advances made possible by the advent of these capabilities.

Acknowledgments
This work was supported in part by the BioEnergy Science Center (BESC). The BioEnergy Science Center is a U.S. Department of Energy Bioenergy Research Center supported by the Office of Biological and Environmental Research in the DOE Office of Science. The work was also supported in part by a DOE—Laboratory Directed Research and Development Fund: Systems Biology 003-00044, and a DOE INCITE award to JCS. Research described was also supported by DOE-OBER grant ERKP704 "Dynamic Visualization of Lignocellulose Degradation" and by the DOE-BER Science Focus Area "Biogeochemical and Molecular Mechanisms Controlling Contaminant Transformation in the Environment" under the Environmental Remediation Science Program. This work was also supported in part by the National Science Foundation (Award No. 0817940 to H. G.).

References

[1] Cosgrove D J. Growth of the plant cell wall. *Nature Reviews Molecular Cell Biology*. 2005 Nov, 6(11):850–61.

[2] Matthews J F, Skopec C E, Mason P E, Zuccato P, Torget R W, Sugiyama J, et al. Computer simulation studies of microcrystalline cellulose i beta. *Carbohydrate Research*. 2006 Jan 341(1):138–52.

[3] Nishiyama Y, Langan P, and Chanzy H. Crystal structure and hydrogen-bonding system in cellulose 1 beta from synchrotron x-ray and neutron fiber diffraction. *Journal of the American Chemical Society*. 2002 Aug, 124(31):9074–82.

[4] Boerjan W, Ralph J, and Baucher M. Lignin biosynthesis. *Annual Review of Plant Biology*. 2003, 54:519–46.

[5] Himmel M E, Ding S Y, Johnson D K, Adney W S, Nimlos M R, Brady J W, et al. Biomass recalcitrance: Engineering plants and enzymes for biofuels production. *Science*. 2007 Feb, 315(5813):804–7.

[6] Ragauskas A J, Williams C K, Davison B H, Britovsek G, Cairney J, Eckert C A, et al. The path forward for biofuels and biomaterials. *Science*. 2006 Jan, 311(5760):484–9.

[7] Gray K A, Zhao L S, and Emptage M. Bioethanol. *Current Opinion in Chemical Biology*. 2006 Apr, 10(2):141–6.

[8] Petridis L, and Smith J C. A molecular mechanics force field for lignin. *Journal of Computational Chemistry*. 2009 Feb, 30(3):457–67.

[9] Kuttel M, Brady J W, and Naidoo K J. Carbohydrate solution simulations: Producing a force field with experimentally consistent primary alcohol rotational frequencies and populations. *Journal of Computational Chemistry*. 2002 Oct, 23(13):1236–43.

[10] Brooks B R, Bruccoleri R E, Olafson B D, States D J, Swaminathan S, and Karplus M. Charmm—a program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*. [Article]. 1983, 4(2):187–217.

[11] MacKerell A D, Bashford D, Bellott M, Dunbrack R L, Evanseck J D, Field M J, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *Journal of Physical Chemistry B*. 1998 Apr, 102(18):3586–616.

[12] Vorobyov I, Anisimov V M, Greene S, Venable R M, Moser A, Pastor R W, et al. Additive and classical drude polarizable force fields for linear and cyclic ethers. *Journal of Chemical Theory and Computation*. 2007 May–Jun, 3(3):1120–33.

[13] Vaiana A C, Cournia Z, Costescu I B, and Smith J C. Afmm: A molecular mechanics force field vibrational parametrization program. *Computer Physics Communications*. 2005 Apr, 167(1):34–42.

[14] Yan J F, Pla F, Kondo R, Dolk M, and McCarthy J L. Lignin 21. Depolymerization by bond-cleavage reactions and degelation. *Macromolecules*. 1984, 17(10):2137–42.

[15] Ralph J, Peng J P, Lu F C, Hatfield R D, and Helm R F. Are lignins optically active? *Journal of Agricultural and Food Chemistry*. 1999 Aug, 47(8):2991–6.

[16] Nishiyama Y, Sugiyama J, Chanzy H, and Langan P. Crystal structure and hydrogen bonding system in cellulose 1(alpha), from synchrotron x-ray and neutron fiber diffraction. *Journal of the American Chemical Society*. 2003 Nov, 125(47):14300–6.

[17] Yui T, and Hayashi S. Molecular dynamics simulations of solvated crystal models of cellulose i-alpha and iiii. *Biomacromolecules*. 2007 Mar, 8(3):817–24.

[18] Bayer E A, Lamed R, and Himmel M E. The potential of cellulases and cellulosomes for cellulosic waste management. *Curr. Opin. Biotechnol.* 2007 Jun, 18(3):237–45.

[19] Bayer E A, Shimon L J, Shoham Y, and Lamed R. Cellulosomes-structure and ultrastructure. *J. Struct. Biol.* 1998 Dec 15, 124(2–3):221–34.

[20] Carvalho A L, Dias F M, Prates J A, Nagy T, Gilbert H J, Davies G J, et al. Cellulosome assembly revealed by the crystal structure of the cohesin-dockerin complex. *Proc. Natl. Acad. Sci. U.S.A.* 2003 Nov 25, 100(24):13809–14.

[21] Henin J, and Chipot C. Overcoming free energy barriers using unconstrained molecular dynamics simulations. *J. Chem. Phys.* 2004 Aug 15, 121(7):2904–14.
[22] Xu J, Crowley M F, and Smith J C. Building a foundation for structure-based cellulosome design for cellulosic ethanol: Insight into cohesin-dockerin complexation from computer simulation. *Protein Sci*. 2009 May, 18(5):949–59.
[23] Phillips J C, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, et al. Scalable molecular dynamics with namd. *J. Comput. Chem.* 2005 Dec, 26(16):1781–802.
[24] Bayer E A, Belaich J P, Shoham Y, and Lamed R. The cellulosomes: Multienzyme machines for degradation of plant cell wall polysaccharides. *Annu Rev Microbiol*. 2004, 58:521–54.
[25] Vocadlo D J, and Davies G J. Mechanistic insights into glycosidase chemistry. *Curr Opin Chem Biol*. 2008 Oct, 12(5):539–55.
[26] Demain A L, Newcomb M, and Wu J H D. Cellulase, clostridia, and ethanol. *Microbiol Mol Biol R*. 2005 Mar, 69(1):124–54.
[27] Clarke A J. Biodegradation of cellulose: Enzymology and biotechnology. Lancaster, PA: CRC Press; 1996.
[28] Rye C S, and Withers S G. Glycosidase mechanisms. *Curr Opin Chem Biol*. 2000 Oct, 4(5):573–80.
[29] Zechel D L, and Withers S G. Glycosidase mechanisms: Anatomy of a finely tuned catalyst. *Accounts Chem Res*. 2000 Jan, 33(1):11–8.
[30] Vasella A, Davies G J, and Bohm M. Glycosidase mechanisms. *Curr Opin Chem Biol*. 2002 Oct, 6(5):619–29.
[31] Berman H M, Battistuz T, Bhat T N, Bluhm W F, Bourne P E, Burkhardt K, et al. The protein data bank. *Acta Crystallogr D*. 2002 Jun, 58:899–907.
[32] Warshel A, and Levitt M. Theoretical studies of enzymic reactions - dielectric, electrostatic and steric stabilization of carbonium-ion in reaction of lysozyme. *J Mol Biol*. 1976, 103(2):227–49.
[33] Sandgren M, Berglund G I, Shaw A, Stahlberg J, Kenne L, Desmet T, et al. Crystal complex structures reveal how substrate is bound in the –4 to the +2 binding sites of humicola grisea cel12a. *J Mol Biol*. 2004 Oct 1, 342(5):1505–17.
[34] Biarnes X, Nieto J, Planas A, and Rovira C. Substrate distortion in the michaelis complex of bacillus 1,3–1,4-beta-glucanase - insight from first principles molecular dynamics simulations. *J Biol Chem*. 2006 Jan 20, 281(3):1432–41.
[35] Guerin D M A, Lascombe M B, Costabel M, Souchon H, Lamzin V, Beguin P, et al. Atomic (0.94 angstrom) resolution structure of an inverting glycosidase in complex with substrate. *J Mol Biol*. 2002 Mar 8, 316(5):1061–9.
[36] Barkay T, Miller S M, and Summers A O. Bacterial mercury resistance from atoms to ecosystems. *Fems Microbiology Reviews*. 2003 Jun, 27(2–3):355–84.
[37] Summers A O. Untwist and shout: A heavy metal-responsive transcriptional regulator. *Journal of Bacteriology*. 1992, 174:3097–101.
[38] Brown N L, Stoyanov J V, Kidd S P, and Hobman J L. The merr family of transcriptional regulators. *FEMS Microbiology Reviews*. 2003, 27:145–63.
[39] Utschig L M, Bryson J W, and O'Halloran T V. Mercury-199 nmr of the metal receptor-site in merr and its protein-DNA complex. *Science*. 1995, 268(5209):380–5.
[40] Song L, Teng Q, Phillips R S, Brewer J M, and Summers A O. ^{19}f-nmr reveals metal and operator-induced allostery in merr. *Journal of Molecular Biology*. 2007, 371:79–92.
[41] Heldwein E E Z, and Brennan R G. Crystal stucture of the transcription activator bmrr bound to DNA and a drug. *Nature*. 2001, 409:378–82.
[42] Ansari A Z, Bradner J E, and O'Halloran T V. DNA-bend modulation in a repressor-to-activator switching mechanism. *Nature*. 1995, 374:371–5.
[43] Morel F M M, Kraepiel A M L, and Amyot M. The chemical cycle and bioaccumulation of mercury. *Annual Reviews of Ecological Systems*. 1998, 29:543–66.

[44] Begley T P, Walts A E, and Walsh C T. Bacterial organomercurial lyase - overproduction, isolation, and characterization. *Biochemistry*. 1986 Nov, 25(22):7186–92.
[45] Pitts K E, and Summers A O. The roles of thiols in the bacterial organomercurial lyase (merb). *Biochemistry*. 2002 Aug, 41(32):10287–96.
[46] Lafrance-Vanasse J, Lefebvre M, Di Lello P, Sygusch J, and Omichinski J G. Crystal structures of the organomercurial lyase merb in its free and mercury-bound forms insights into the mechanism of methylmercury degradation. *Journal of Biological Chemistry*. 2009 Jan, 284(2):938–44.
[47] Gopinath E, and Bruice T C. Assistance of protodemercuration by bis-thiol ligation and nucleophilic catalysis - a model study which relates to the organomercurial lyase reaction. *Journal of the American Chemical Society*. 1987 Dec, 109(25):7903–5.
[48] Begley T P, Walts A E, and Walsh C T. Mechanistic studies of a protonolytic organomercurial cleaving enzyme—bacterial organomercurial lyase. *Biochemistry*. [Article]. 1986 Nov, 25(22):7192–200.
[49] Melnick J G, and Parkin G. Cleaving mercury-alkyl bonds: A functional model for mercury detoxification by merb. *Science*. 2007 Jul, 317(5835):225–7.
[50] Strasdeit H. Mercury-alkyl bond cleavage based on organomercury lyase. *Angewandte Chemie-International Edition*. 2008, 47(5):828–30.
[51] Miller S M. Cleaving c-hg bonds: Two thiolates are better than one. *Nature Chemical Biology*. 2007 Sep, 3(9):537–8.
[52] Parr R G, and Yang W T. Density functional theory of atoms and molecules, international series of monographs on chemistry, vol. 16. New York: Oxford University Press, 1989.
[53] Brooks C L I, Karplus M, and Pettitt B M. Proteins: A theoretical perspective of dynamics, structure and thermodynamics. New York: Wiley, 1988.
[54] Brown K G, Small E W, Peticola.Wl, and Erfurth S C. Conformationally dependent low-frequency motions of proteins by laser raman spectroscopy. *Proceedings of the National Academy of Sciences of the United States of America*. 1972, 69(6):1467–&.
[55] Daniel R M, Dunn R V, Finney J L, and Smith J C. The role of dynamics in enzyme activity. *Annual Review of Biophysics and Biomolecular Structure*. 2003, 32:69–92.
[56] Daniel R M, Finney J L, Reat V, Dunn R, Ferrand M, and Smith J C. Enzyme dynamics and activity: Time-scale dependence of dynamical transitions in glutamate dehydrogenase solution. *Biophysical Journal*. 1999 Oct, 77(4):2184–90.
[57] Daniel R M, Smith J C, Ferrand M, Hery S, Dunn R, and Finney J L. Enzyme activity below the dynamical transition at 220 k. *Biophysical Journal*. 1998 Nov, 75(5):2504–7.
[58] Diehl M, Doster W, Petry W, and Schober H. Water-coupled low-frequency modes of myoglobin and lysozyme observed by inelastic neutron scattering. *Biophysical Journal*. 1997 Nov, 73(5):2726–32.
[59] Doster W, Cusack S, and Petry W. Dynamical transition of myoglobin revealed by inelastic neutron-scattering. *Nature*. 1989 Feb 23, 337(6209):754–6.
[60] Doster W, Cusack S, and Petry W. Dynamic instability of liquid-like motions in a globular protein observed by inelastic neutron-scattering. *Physical Review Letters*. 1990 Aug 20, 65(8):1080–3.
[61] Hayward J A, Finney J L, Daniel R M, and Smith J C. Molecular dynamics decomposition of temperature-dependent elastic neutron scattering by a protein solution. *Biophysical Journal*. 2003 Aug, 85(2):679–85.
[62] Hayward J A, and Smith J C. Temperature dependence of protein dynamics: Computer simulation analysis of neutron scattering properties. *Biophysical Journal*. 2002 Mar, 82(3):1216–25.
[63] Knapp E W, Fischer S F, and Parak F. The influence of protein dynamics on mossbauer-spectra. *Journal of Chemical Physics*. 1983, 78(7):4701–11.

[64] Kurkal V, Daniel R M, Finney J L, Tehei M, Dunn R V, and Smith J C. Enzyme activity and flexibility at very low hydration. *Biophysical Journal*. 2005 Aug, 89(2):1282–7.
[65] Reat V, Dunn R, Ferrand M, Finney J L, Daniel P M, and Smith J C. Solvent dependence of dynamic transitions in protein solutions. *Proceedings of the National Academy of Sciences of the United States of America*. 2000 Aug 29, 97(18):9961–6.
[66] Reat V, Patzelt H, Ferrand M, Pfister C, Oesterhelt D, and Zaccai G. Dynamics of different functional parts of bacteriorhodopsin: H-h-2 labeling and neutron scattering. *Proceedings of the National Academy of Sciences of the United States of America*. 1998 Apr 28, 95(9):4970–5.
[67] Roh J H, Curtis J E, Azzam S, Novikov V N, Peral I, Chowdhuri Z, et al. Influence of hydration on the dynamics of lysozyme. *Biophysical Journal*. 2006 Oct, 91(7):2573–88.
[68] Roh J H, Novikov V N, Gregory R B, Curtis J E, Chowdhuri Z, and Sokolov A P. Onsets of anharmonicity in protein dynamics. *Physical Review Letters*. 2005 Jul 15, 95(3):–.
[69] Tournier A L, and Smith J C. Principal components of the protein dynamical transition. *Physical Review Letters*. 2003 Nov 14, 91(20):–.
[70] Tournier A L, Xu J C, and Smith J C. Translational hydration water dynamics drives the protein glass transition. *Biophysical Journal*. 2003 Sep, 85(3):1871–5.
[71] Bee M. Quasielastic neutron scattering: Principles and applications in solid state chemistry, biology and materials science. Philadelphia: Adam Hilger, 1988.
[72] Darden T, York D, and Pedersen L. Particle mesh ewald—an n.Log(n) method for ewald sums in large systems. *Journal of Chemical Physics*. 1993 Jun, 98(12):10089–92.
[73] Essmann U, Perera L, Berkowitz M L, Darden T, Lee H, and Pedersen L G. A smooth particle mesh ewald method. *Journal of Chemical Physics*. 1995 Nov, 103(19):8577–93.
[74] Tironi I G, Sperb R, Smith P E, and Vangunsteren W F. A generalized reaction field method for molecular dynamics simulations. *Journal of Chemical Physics*. 1995 Apr, 102(13):5451–9.
[75] Schulz R, Lindner B, Petridis L, and Smith J C. Scaling of Multimillion-Atom Biological Molecular Dynamics Simulation on a Petascale Supercomputer, *Journal of Chemical Theory and Computation*, 2009, 5(10):2798–2808.

From grid to cloud, the STAR experience

Jérôme Lauret,[1] Matthew Walker,[2] Sebastien Goasguen,[3] and Levente Hajdu[1]

[1]Brookhaven National Laboratory
[2]Massachusetts Institute of Technology
[3]Clemson University

Abstract. In recent years, Cloud computing has become a very attractive paradigm and popular model for accessing distributed resources. The Cloud has emerged as the next big trend after the so-called Grid computing approach. The burst of platforms and projects providing Cloud resources and interfaces at the very same time that Grid projects are entering a production phase in their life cycle has, however, raised the question of the best approach to handling distributed resources. Especially, are Cloud resources scaling at the levels shown by Grids? Are they performing at the same level? Can they be compared? What are their overhead on the IT teams and infrastructure? Since its first use of Cloud resources on Amazon EC2 in 2008/2009 using a Nimbus/EC2 interface, the STAR experiment software team has tested and experimented with many novel approaches: from a traditional native EC2 approach to the Virtual Organization Cluster (VOC) at Clemson university and Condor/VM on the GLOW resources, STAR has ramped up the job scale step by step to achieve stable operation at the 1,000 jobs level. This paper presents an overview of our findings and reports on practical usage of truly opportunistic use of resources.

1. Introduction

Faced with huge data challenges, High Energy and Nuclear Physics experiments and other communities have worked toward the design and engineering of new computing models and paradigms where resources at given sites are seen as part of a global infrastructure. At a higher level, the aim was to present the resources as one virtual entity and thus provide a consistency to the user who may interact with a global pool of resources and not deal with the details. The era of Grid computing was born, and since its inception it has shown its success in data processing and handling over vast distance and across many continents. Grid projects such as the Open Science Grid matured and hardened the technology to provide national grids on which Virtual Organizations (VO) could harvest vast amounts of supposedly widely shared resources. However, the many limitations of Grids were also obvious from the start—with heterogeneity at the heart of its philosophy and the plethora of environments, the exploitation of such global resources became rather challenging: experiment's team spreading thin, the support of complex software stack made the opportunistic usage of sparse and unused resources nearly impossible, and most VOs reverted to the use of pre-installed software with local software support and maintenance. This "modi operandi" allowed minimizing internal cost as the vast amount of available combinations between software, middleware, compilers, libraries and event services would imply a large combination and support platform and hence, a growth in the support team, not to mention the effort in quality control and assurance needed for experimental teams having reached peak productivity. The arrival of Cloud computing has introduced a new reality and feasibility beyond proof of principles: relying on virtualization, a virtual machine can contain all the required components from a tailored and specific operating system (OS, up to the minor revision if need be) to

a specific middleware to an experimental software stacks pre-verified, controlled, and approved at their home institution via regression test suites. The burden on the experimental team suddenly appeared to be normalized to their common environment, making the dream of harvesting unused resources "on-the-fly" possible.

However, while Cloud computing models are rapidly reaching maturity, unlike Grids, they do not benefit from a unified approach and standard interface, and many models are available on the market from private commercial clouds to public national laboratory or university community clouds. This paper will present a few models tested by the RHIC/STAR experiment as well as our usage experience.

1.1. The RHIC/STAR Experiment's Data Challenge

The Solenoidal Track At RHIC (STAR) experiment is a Nuclear Physics experiment located at the Brookhaven National Laboratory (BNL) and part of the RHIC program. At STAR, physicists and skilled specialists from across the world are working hard to understand the nature of the early universe and the tiniest building blocks of matter through the study of nuclear collisions at the highest energies achieved in the laboratory.

To achieve this goal, the STAR experiment has developed an ambitious program now entering the Petascale data challenge (Figure 1). In this resource planning, it was shown that only two passes over the raw data and one pass user analysis was possible within the current funding profile; another portion of user analysis as well as all resources to provide simulation capabilities needed to be outsourced and come from other sources than the RHIC funding. Also, with the experiment moving from a statistically challenged data sample to large data sample, it enters an era where systematic uncertainties are predominant and hence, raise concerns on possible (unforeseen) needs for additional simulation for understanding its effect on the detector response, fine grain efficiency, and momentum corrections. Such a simulation will be discussed in Section 0, constituting a real need for truly opportunistic resources.

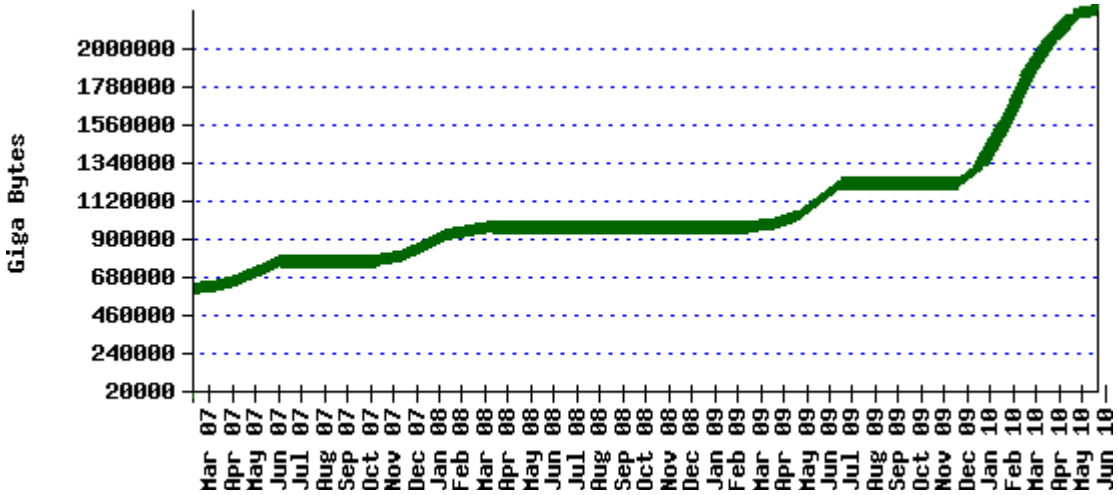


Figure 1. STAR raw data amount stored into archival Mass Storage as a function of time. The curve is restricted to the range 2007 to 2010; the last year, also known as Run 10, has accumulated nearly as much data than all previous years combined with a total of 2.2 PBytes of data.

To face this challenge, STAR has had a long-standing participation in Grid co-laboratories programs. Since its early phase, STAR has been part of the Particle Physics Data Grid consortium, the Trillium project and its next incarnation, the Open Science Grid. But in parallel, and facing early challenges in porting large application in Physics production mode to heterogeneous platforms, the

STAR experiment Software and Computing team have also carried and maintained active participation and testing in alternative distributed computing approaches [8,9,10] with its first truly last minute opportunistic use of Cloud computing based resources in early 2009. What has driven the experiment to outreach to other distributed computing models? What are the limitations of grids for experimental groups ready to harvest or aggregate their resources?

1.2. Grids, Success and Limitations—a Problem Analysis

Grids have been successful at aggregating geographically separated computing and storage resources (at sites) and presenting to their users and communities an aggregate or federated "cluster" like component. On such federated resources, the breadth of activities has brought their first benefits, starting from simply enabling data movement across the network and cyber-infrastructure [11,12,13] and continuing to the development and/or the hardening of middleware and infrastructure, leading to reaching an ever-increasing operational sustainability. From 65% success rate in 2006, 85% in 2006–2007, 90% in 2009 and greater than 97% efficiency on job success using the OSG software stack, the deliverables and benefits are undeniable to first sight and the resource aggregation strategy paid off after long years of efforts. In detail, the STAR experiment seldom uses Grid resources and at the level of 64,000 jobs a week and mostly using dedicated sites that is, sites where the STAR software is pre-installed and maintained locally. This could be easily understood by realizing that STAR, after nearly 10 years of productivity and refinement of its software developed by hundreds of scientists, has reached a point where the stack is composed of 2.5 million line of codes, depends on multiple external libraries (e.g., MySQL, XML, ROOT) and relies on a mix of compilers and languages (from C, C++ and FORtran) requiring Linux core support for those languages (e.g., libstdc++, libg2c). The heterogeneity of Grids does not ensure the existence of such components and compiling "on the fly" is already a no-go due to the lack of existence of compilers. In addition, there are no information systems providing enough information allowing for safely choosing sites fulfilling the proper conditions, and as a consequence, experimental groups have reverted to "pilot jobs" where a simple script runs on the remote resource and checks first the availability of components before downloading and installing the VO software stack and an appropriate payload. Additionally, Grids are very cryptic—troubleshooting is still far from adequate and upon errors, messages from the stack are simply not understandable by the average user (some error may have many meanings requiring investigation). As a result, bursts of resources are hard to acquire and in fact, inspection of other experimental projects reveals little to no opportunistic use of Grid resources and an operation only sustainable to the extent a "support team" exists at these facilities.

In contrast, and while clouds do not offer solutions for adequate monitoring and troubleshooting and cannot be sold as "silver bullets," virtualization offers a "software container" based approach and therefore provides definite advantages while running over heterogeneous resources: the combination of OS, libraries, software stack version is not only under full control of the experiment, but the entire combination can be pre-validated via regression test suite. This in turn saves resources, both human and computing cycles (although the human cost is larger in this case) which otherwise would need to be spent to ensure Physics reproducibility and accuracy over heterogeneous environments. As a bonus, the prepared VM can be archived and serve as a software container repository to ensure later checks of past results or cross comparisons of results obtained with older software often not supportable on newer machines (and newer compilers) without additional porting efforts.

A good question would be whether the Cloud simply displaces the problem: while the software and environment can be preserved, the VM technologies are rapidly evolving and it becomes questionable if today's VM formats would at all be compatible with tomorrow's new virtualization approaches. However, hopes remain as not only tools such as rBuilder are being developed in the private sector to address release life-cycles (as well as long term maintainability of a virtualized on-demand platform provisioning approach), but solutions such sa Xen VM or VirtualBox provide tools to convert from the most common VM formats to their native image format. It is expected that this trend will continue as the need for conversion is already recognized. Certainly, the many advantages of Clouds have been

noted by an enthusiastic community in search for true solutions and, as shown on Figure 2, the interest in Cloud now surpasses that of Grid computing.

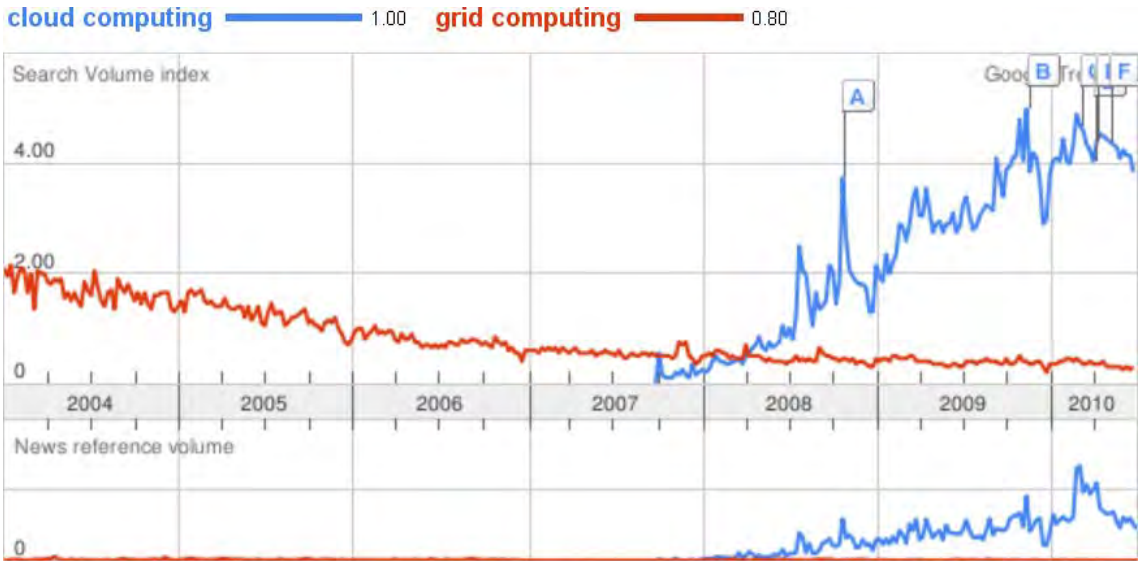


Figure 2. Google search trends for Cloud and grid computing terms. The decrease of interest for grid computing and increase for cloud computing is indicative of a growing community (at large) interest for cloud computing now surpassing Grid computing. The switch of interest happened in 2008.

2. Clouds, Virtualization Models and STAR Tests

2.1. Cloud Key Features and Clouds

We do not intend to describe the anatomy of Clouds in great detail. It is, however, noteworthy to mention that while Grids ask for job slots on Worker Nodes (WN), cloud interfaces are intrinsically requesting VM instances to start on a common infrastructure (regardless of the nature of its "flavor"). In our work, the notions of Software as a Service (SaaS), Platform as a Service (PaaS), or Infrastructure as a Service (IaaS) which could be combined together is not relevant, those definitions and all related notions could have been defined and explained by NIST—the IaaS runs our fully provided and customized VM containing all software provisioning we will need to run and establish a simulation, calibration, real data reconstruction or user analysis.

In this paper, we focus on a Nuclear Physics experiment workflow and its feasibility on the grid: whether or not little or large inputs are needed, whether or not external meta-data (database) information is needed for an event reconstruction producing Physics analysis usable quantities, our workflow does produce a significant amount of output (could be as large as several GB per job) and network transfer in/out of Cloud into a central archival storage system at BNL (For science preservation) was needed to conclude on usability.

In Figure 3, we present a high level view of access to resources through Grid and Cloud computing. The front-end to users presents itself as a thin layer allowing access to the compute elements and often involves a Local Resource Manager System (LRM or LRMS are often standard batch systems)—as we previously noted, while Grids provide access and scheduling of jobs on a "raw" worker nodes, Clouds instantiate Virtual Machines on worker nodes. Cloud interfaces may present themselves as simple Web front ends to more complex command line grid-like job interface. All tested model fit within this representation and conceptualization, what changes is the domain or boundary for virtualization.

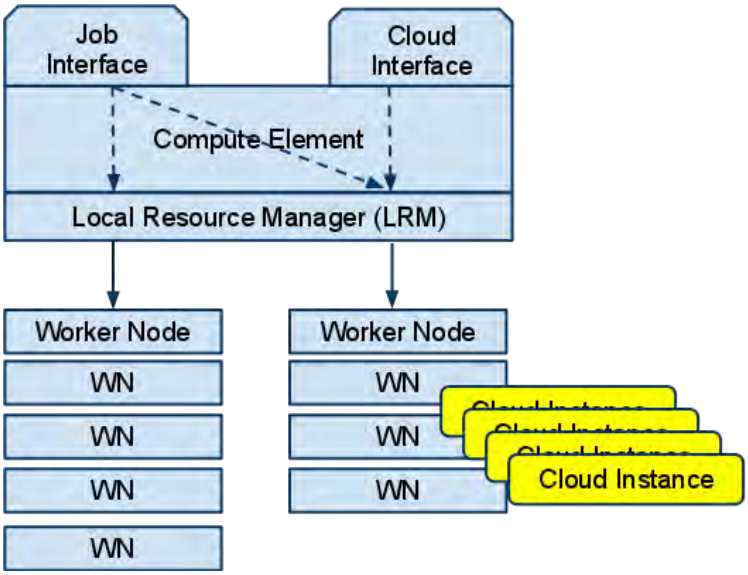


Figure 3. High level representation of access through a Grid or a Cloud interface.

2.2. Virtualization Models Testing, Pro and Cons

Many cloud providers currently exist, and we restricted our testing to five models: (1) Amazon/EC2 native platform a pure Web based front end, (2) a combination of the Nimbus project and Amazon/EC2, (3) The Clemson University Virtual Organization (VOC) model [20,21], (4) The Condor/VM model , and (5) The Kestrel model . Due to the different nature of those approaches, the flexibility on the clusters (existing communication in and out of the virtualized world, ability to monitor components outside the VM or not), we could not come up with a consistent set of standard monitoring and test suite, but rather tried to reach the same conclusion using different methods that we will describe in detail in the appropriate sections.

2.2.1. Amazon EC2

In the native Amazon/EC2 approach, and while command line tools are available, the interaction with, and monitoring of, the system is mainly Web based. The general communication workflow is represented on Figure 4. Amazon EC2 uses a custom image format that can be created from any system using a variety of methods. In this exercise, a VMWare image with Ubuntu and the STAR libraries installed was converted to an Amazon Machine Image (AMI) using the EC2 AMI Tools, which are command-line utilities that help create an image and upload it to Amazon S3. The AMI was deployed using the standard Amazon Web Services (AWS) interface to EC2. Additionally, secure shell custom keys were put inside the image allowing direct connection from the outside in. Data transfer out was handled using either temporary ssh-key based transfer to a low security requirement storage site or using a more secure grid-proxy using myproxy delegation mechanism to a secured site. The system as it stands is not however self-sufficient in the sense the Amazon/EC2 interface allows starting, stopping VMs but does not allow, by itself, to create a virtual cluster equipped with its own RMS (local or global) a-la-Grid gatekeeper. To circumvent this problem, each VM instance connected on startup to a jobs server to receive job parameters, the workflow was a simple Monte-Carlo simulation requiring only a few parameter variation (and statistics were built for each set using a different seed per job). After completing the job and transferring the results back to the main facility at BNL, each instance shutdown.

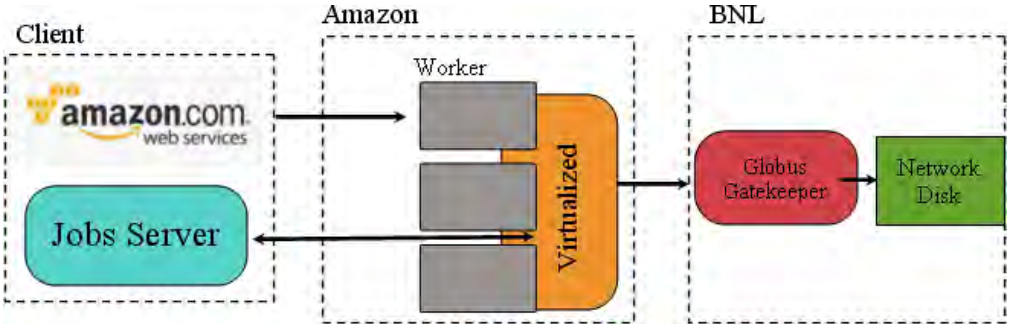


Figure 4. Architecture diagram of the EC2 interface. In this model, images are selected from the Amazon Web Services (AWS) interface and instantiated on the EC2. The images connect to the job server on startup, retrieve any necessary specifications, run their jobs, transfer the results back to the grid, and shut down.

A first observation is that since we have no access to the underlying infrastructure, the knowledge of efficiency is biased in the sense that whatever we asked as VM started (but we cannot differentiate between an over-commitment of VM "under the hood" or the start and death of some VM affecting farm occupancy). However, once the VMs were started and stable, we observed a 99% efficiency on a 100 VMs order of magnitude. The IO out of each WN was of the order of 5 MB/sec and we did not perform a scaling beyond a few tens of simultaneous transfers from multiple VMs (this data transfer sufficed for our exercise). Performance comparisons were also made between the various types of instances available: using the same program flow (using no more than 1 GB of memory), we tested multiple "instances" namely, the small, medium and large instance. The results were surprising: on a small instance, the ratio of CPU/clock time indicated a 40% CPU usage efficiency (at 0.085$/hour, this leads to a 0.21$/hour effective pricing) while on the medium instance, one of the same program would show 99% CPU efficiency (similar to the large instance performance but costing 0.21$/hour). We also observed that within one medium instance, we could in fact start our program twice in parallel, leading to an effective pricing of 0.09$/hour. Both Medium and Large instances were able to achieve approximately a factor of four increase in throughput compared to small instances.

The main benefit of Amazon/EC2 is its simplicity, its pay as you go feature but also its concept of VM repository (AMI/appliance). This concept of a repository of approved images would be needed for experimental group so the flow of random and un-checked software stack would be prevented. The pricing seem competitive to us as well: at this price, a 100 jobs week long simulation would cost ~$1,510 (without storage) and a year long would be 79k$ all included (that is, no IT to maintain a cluster, no additional electricity or cooling bill). For university resources, we feel this cost is affordable and may well provide burst of resources required by experimental groups. On the down side, Amazon/EC2 security model has nothing to offer compared to the AAA features of the grid infrastructure (one may even qualify it as weak).

2.2.2. Amazon EC2+Nimbus

The Nimbus/EC2 model is similar to the previous model with slight twists and important improvements. The first important change is that not only VMs are started but they also started two different images: one acting as a standard Grid gatekeeper (with the full OSG software stack deployed "within") and the others (the WNs), having a standard OSS client software stack in addition to the STAR required software + a batch system client registering to a Master running on the Gatekeeper (Grid server software stack) as well as containing a LRMS for batch management. The Worker Nodes' batch clients subscribe to the gatekeeper and the combination of gatekeeper + WNs appears as a virtual cluster. In this approach, the resource appears as a virtual cluster and to the external user, is simply yet another site as part of federated set of resources as represented on Figure 5. In fact, STAR has used its standard Meta-scheduler (SUMS) with plugins to submit jobs in the most standard way for Grid operations while Nimbus allowed easily configuring and starting the virtual cluster.

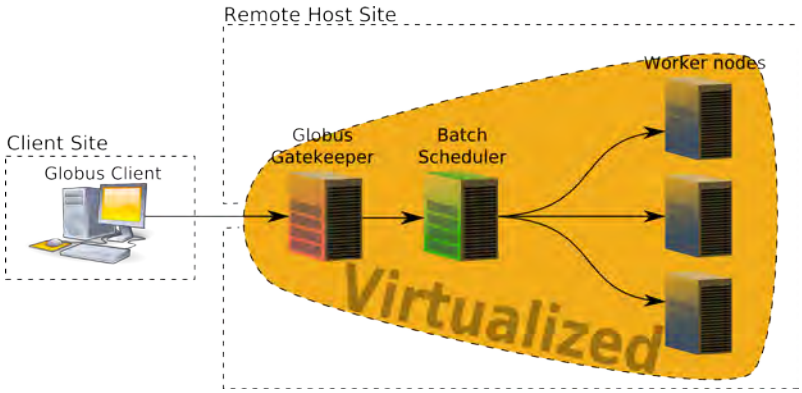


Figure 5. Architecture diagram of the EC2+Nimbus interface. In this model, a VM is started acting as a grid gatekeeper wile Worker Node (WNs) are started to handle the main processing. GK+WNs effectively form a standard Grid (virtualized) cluster and resource.

To first order, this approach was much simpler of use for adding resources to our Grid operations pool. We scaled to a few hundred VMs but the efficiency of running job after the VM were instantiated dropped slightly to 85% efficiency on first submission, reaching 97% efficiency on second submission of the failed jobs. The drop was mainly due to two factors: a scalability issue in the version of the LRM (PBS in our case) which would drop communication between WN and the Master losing a fraction of the jobs and the interaction between the Grid job submission component (a.k.a. Condor-G) and PBS. After tuning and patching however, the final achieved efficiency is comparable to the one we can observe on the OSG native infrastructure as noted in Section 0 and the benefits drawn from having a way to easily start/tear down virtual clusters on cloud resources tend to overcome the small loss. A con is that there is a delay between starting a gatekeeper and the WNs: this delay is due to the first identified problem of Cloud—as soon as the model become more complex, virtual machines need a contextualization that is, a mechanism to at least import a-posteriori information within the instance.

In our case, the IP of the master nodes is needed by the client batch system. Nonetheless, the testing and exploitation of such mechanism has allowed STAR to make truly breakthrough real-life data production run in 2009 as reported in reference . This was to our knowledge the very first real usage of Cloud computing resources with results used in an international conference (Quark Matter 2009 in this case).

2.2.3. The Virtual Organization Cluster (VOC) model at Clemson

The VOC model has been described in great length in references [20,21] and is represented on Figure 6. The user interacts and submits jobs to a Globus gatekeeper using standard grid tools; the VOC system and daemons are responsible for dynamically starting (or shutting down) an appropriate number of VMs on the WN to satisfy both demand and resource allocation for a given VO. Note that the switch between native resource to virtualized resources was solely based on authenticating as a member of the STAR VO but other more complex mechanisms could be defined, following this general philosophy. Upon instantiation, a LRM “client” subscribes to the master as in the Nimbus/EC2 case. However, the virtualized domain is only the set of VMs (the gatekeeper may not be virtualized as in the previous model and may share Cloud and Grid submission). The publish/subscribe mechanism in this case provided ultimate transparency—from a job submitter stand point, there are no other differences in submission comparing to a standard Grid submission; the back-end VM ensures proper software provisioning and the user is completely agnostic to the technology used behind the scene.

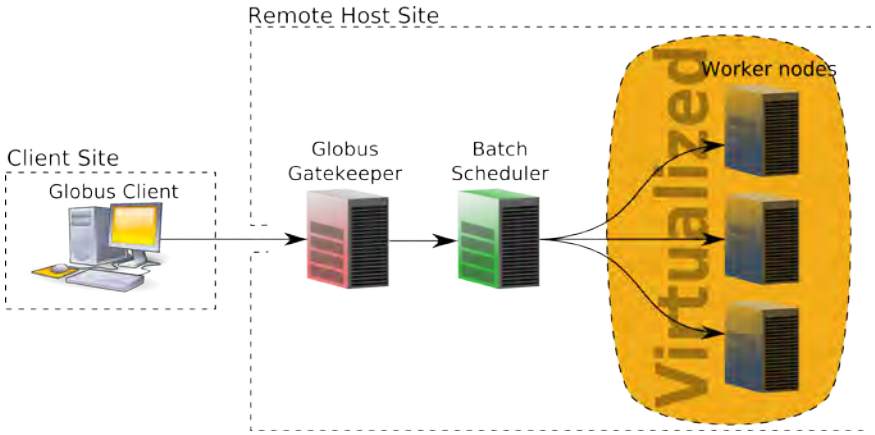


Figure 6. The Virtual Organization Cluster (VOC) model presents itself as a standard Grid job interface while the back-end is virtualized.

Our testing was rather limited: from a few tens of VMs, the results can hardly be compared to the other models but rather, features could be extracted. Not surprisingly, the job efficiency (including in this case the ration of requested versus started VMs) was 100%. In this model the contextualization remains to some extent a site specific overhead but, since we made use of KVM , we minimized its burden by heavily relying on the snapshot feature KVM provides. This feature allows re-directing all changes done inside the VM to a local storage hence bypassing the need to mount a specific storage element and reducing local network IO overhead by writing over LAN. We also found in this phase of our testing that pre-caching the images on the local node (as possibly achievable) made the difference between a delayed and an immediate response of the demand satisfaction as supplied VM would start. The total overhead between submitting a job to the gatekeeper to having the job start within the VMs for an extended period of time and different load scenari showed the overhead to be less than 1% of the total workflow lifespan.

2.2.4. Condor/VM

The Condor/VM model is a simple model where the Grid gatekeeper accepts a standard grid job description but drops all directives but the one requesting to start a VM instance. The system accepts a VMWare image that is then submitted to the Condor cluster as a job. In this mode, a similar model to the Amazon/EC2 was adopted as Condor/VM is also not self-sufficient: after startup, one cannot communicate with the VM and hence, a mechanism to pull a job in has to be supplied (a similar mechanism as the one described in Section 0 was used). Figure 7 represents the communication workflow in this model. All VMs were configured via NAT and had a connection from inside out, allowing transferring, as before, the data out from the cluster to the main STAR facility at BNL. The amount of contextualization in this case is minimal (no batch system to configure as the batch system, Condor, leaves outside the VM starting them as needed; mounting file system could be a standard operation; there was no network/DHCP address lease issues). Since Condor controlled the startup of the VM, a possible problem was that the VMs would expire after a certain pre-defined lifetime determined by the local Condor pool configuration. This time was 24 hours in this case and jobs were calculated to fit within this time modulo uncertainty inherent to Monte-Carlo simulations.

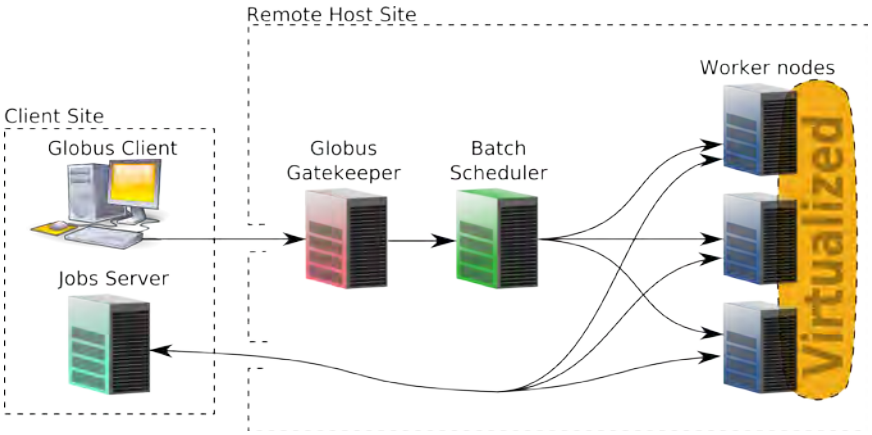


Figure 7. The Condor/VM model is a simple scalable approach whereas the submission to a gatekeeper only requests the startup of VMs onto the cluster. Condor/VM acts as the top layer and controller of the startup and shutdown of VMs.

Over a scale of 500 VMs, 10% of the VMs never started on demand causing a net loss of slots at the start (but detectable and recoverable), 15% of the VMs stopped unexpectedly (crashed) and 5% of the jobs were killed due to VM lifetime expiration. The global job efficiency is therefore 73% (with peak at 80–85%). It was also necessary to pre-stage the image to all of the WNs before issuing the Condor job request. This reduced the time to start 500 machines from several hours to about an hour. Here again, the pro is that this Cloud presents itself with a standard Grid interface (but works best for those VO who already have a pull model, where an external job server provides the workflow information to the VMs which may be a con) and the setup of such model requires little contextualization which may be standardized across all supported VOs, reducing overhead on local staffing—the system may also support a vast amount of images (they could later be organized into VO specific repositories of images).

2.2.5. The Kestrel model

The Kestrel model describes a method for controlling jobs on images across a variety of sites. VMs were started at the Clemson Palmetto cluster and at CERN using PBS 0. Upon startup, the images connect to the Kestrel Manager, reporting their presence and any attributes that are defined (e.g., which STAR library version). The manager then schedules jobs to workers that satisfy the necessary requirements. Jobs can be submitted to the manager using command line utilities or instant messaging clients that support Jabber . Control of the number of available machines is not yet integrated into Kestrel but a home-made job watcher allowed us to feed the system with jobs. Figure 8 illustrate the response of the system. In red, we represent the number of available machine on a shared resource cluster (the number of machine which may be claimed at a given time) and in green, the number of VMs we could start within minutes. The blue curve represents the number if idle machine—a growth would indicate the Kestrel manager is not able to communicate with the VMs and feed a job. As we can see, a temporary glitch happened on July 25th and need better analysis but overall, the resource availability and our ability to harvest them as opportunity arise is astonishing. The difference between the number of machines started in PBS and those seen by the Kestrel manager on average differed by roughly 1%. The manager was able to utilize newly present machines on the order of seconds, but took on the order of several minutes to notice that a VM had disappeared.

It is important to mention that the usage of the Kestrel Cloud model is beyond an exercise: it represents a complex Monte-Carlo simulation followed by an event reconstruction requiring the full STAR framework to be available within the VM, database included. This simulation is the first use of Cloud for studying detector's systematic effects on DiJet productions. For this effort we added, within

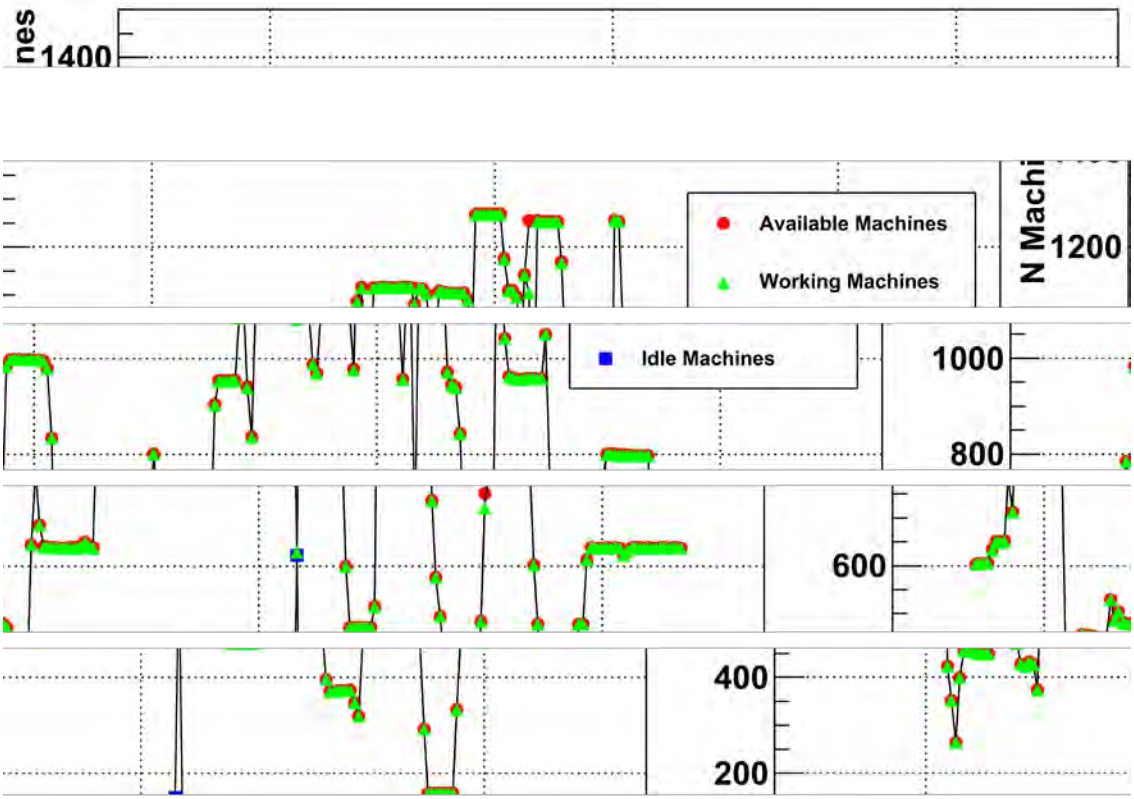


Figure 8. Tracking on the number of jobs as a function of time using the Kestrel system. The number of instantiated VM tracks with the number of available nodes (not used by other demands) indicating a good response of the system overall. A guaranteed allocation of 1,000 slots for a few days around July 21st shows we exceed the number of slots and took advantage of the farm empty slots.

the VM, a MySQL server started upon booting the VM. The snapshot database, a 0.5 GB uncompressed file, is de-compressed upon startup before MySQL starts and since we use the snapshot mode of KVM, this leads to the database to appear local to the worker. Each VM/job then has its own database (hence not scalability issue and no network connection issues). This large simulation has been to date our largest use of Cloud computing model, both in simultaneous number of VMs (reaching the 1,000 for days to spanning already over a month length at 700 VMs average). Nearly 12 billion events have been generated by the Monte Carlo event generator using over 40,000 CPU hours. From these events, over 6.5 TB of data have been transferred back to BNL using grid-ftp, which the VMs are able to access using myproxy .

2.3. Current Status, Projections and Perspectives

2.3.1. Current scale and evolution

Figure 9 summarizes the scale and dates at which we performed the diverse exercises reported in this paper. While we have now reached and demonstrated the level of a 1,000 VMs and project that if this trend persists, the 10,000 to 100,000 levels will be reached by the end of 2010, an operational scalability which may satisfy STAR's need.However, we are to date a long way away from the sustainability provided by the OSG infrastructure, peaking at a near 800 k jobs.

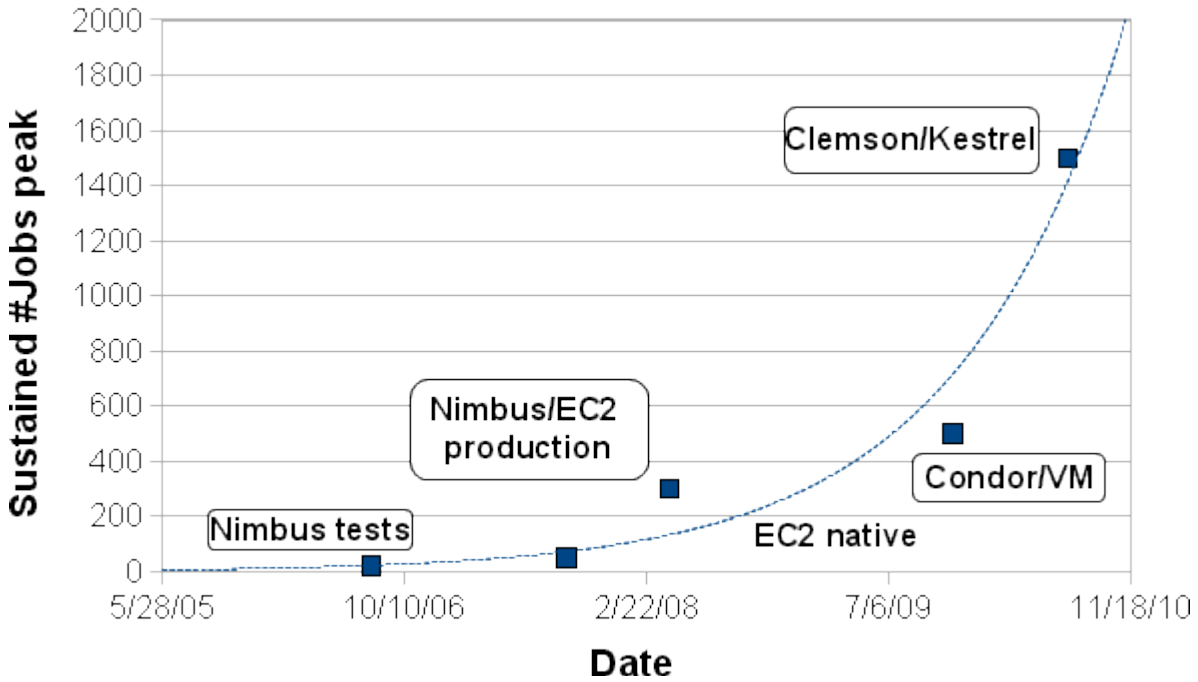


Figure 9. STAR testing scale as a function of time. From the initial Nimbus test in 2006 to the Kestrel model in summer 2010, the scale of sustainable operation has grown by two orders of magnitude, indicating a maturation of the technology and approaches. If this trend persists, we project that within the end of 2010, a number of jobs of the order several 10k to 100k jobs would be feasible and sustainable.

2.3.2. Evolution of Cloud infrastructure, advantages and possible benefits

Nonetheless, Cloud computing shave generated not only a community wide interest but is likely at the peak of expectations . In parallel, the private sector such as Amazon.com, is aggressively addressing the demand of the HPC community by providing offers suited for large scale computational work: new "Cluster compute" instances on EC2 now provide 1.6 TB of storage, 23 GB of memory and 10 Bg network connection to the outside world for a $1.60/hour quoted price. It remains to be seen if this pricing is beneficial but once promises are finally delivered by Cloud and virtualization technology the possibility to really reach the ultimate goal of harvesting opportunistic resource will have arrived. The technology is so mature that several projects have also seen birth over the past few years (DOE Magellan project , NSF Azure Cloud) with aim to evaluate and/or create a national Cloud infrastructure for frontier science. It is to be noted as well that in our testing of Amazon/EC2, the network data transfer seemed inadequately low to also sustain a full data production operation requiring in STAR a total of nearly 3 Gb/sec transfer rate. The Cloud-ification of national laboratories could alleviate this problem by creating Cloud infrastructure on already network provisioned infrastructure. National Laboratories also provide low cost large data storage solution (mass storage, archival storage, disk storage) which we omitted in our price estimates (Amazon S3 is not currently competitive to the cost at National Laboratories for Peta-Scale storage, and lifetime storage guarantees are not yet part of many commercial cloud's real objectives). Several advantages of Cloud and/or virtualization appears to our community.

Beyond National Laboratories, the ability of virtualization to carry along software and environment does not seem beneficial only in a context of Cloud and distributed computing. For a typical embarrassingly parallel community with simple workflows, and as we noted in Section 0, maintenance of a vast variety of platforms is costly. In a typical Tier architecture model of distributed computing,

cost goes beyond Tier-0 centers and span to the Tier-2 support where software installation is often done by a volatile workforce. Easy software provisioning with no need for post-installation validation would serve as an asset to the creation and use of local resources at Tier-2 centers.

The role of Cloud computing in the era of Exa-Scale computing is also to be understood—what used to be named "off the shelf" commodity hardware composed of nodes equipped with a few cores and nearly no parallel processing capabilities may become obsolete as the hype of Exa-scale and the multi-core era ramps up to take on the community's full attention. Cloud would allow sites to drop cost ineffective clusters and fold them onto larger multi-core resources (providing IO and inefficiency are not concerns) changing dramatically the notion of "Clusters"—in this scenario, a "Cluster" may simply be a piece of a larger resource pool provisioned by Cloud technologies.

Cloud interfaces as we have seen remains however heteroclitic and this may reduce the penetration, early adoption or hardening of Cloud technologies in experimental groups. However, several projects are already on the way to provide unified interfaces to the cloud; example of such work is the incubator libcloud project , Delta-Cloud , or StratusLab . A possible high level diagram of how to integrate clouds to the main stream projects such as the OSG is presented in Figure 10. Our experience with Cloud computing shows many approaches already provide (following the discussion in Section 0) plug-and-play or transparent submission features (VOC, Nimbus) while others need to be supplemented by additional components to offer the full versatility the Grid provides. Such a proposal would need to be evaluated within the OSG consortium and may leverage the work and deliverables from the ExTENCI project [39], a joint TeraGrid and OSG project aimed to explore the use of VMs across TeraGrid and OSG test beds.

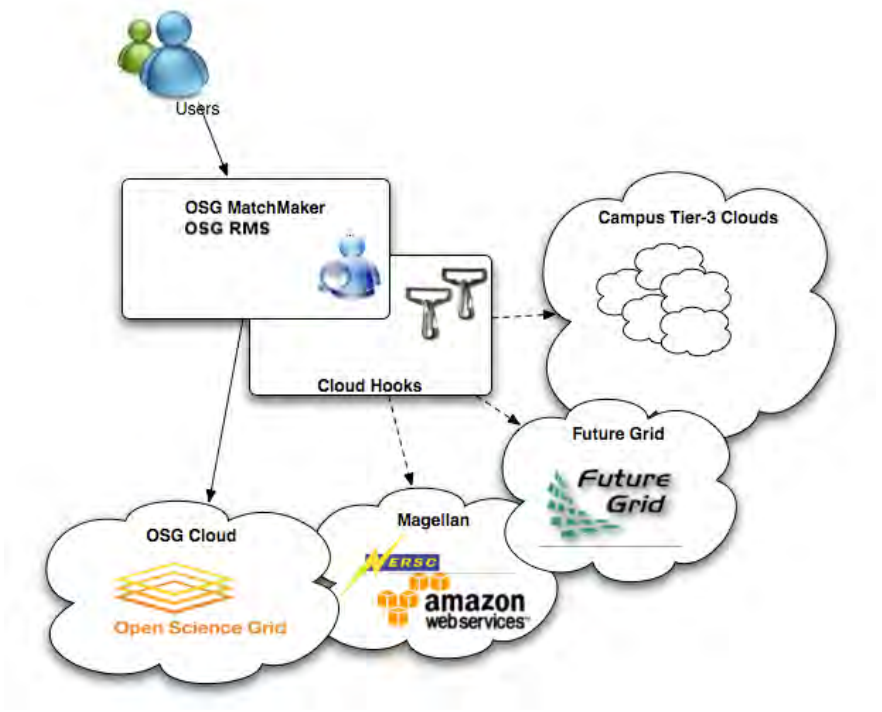


Figure 10. At the highest level of abstraction, a possible interface to the Open Science Grid could be a standard OSG Match Maker or standard GRMS with plug-and-play hooks to Cloud resources. While the Grid layer would act as a standard globus stack, hooks would handle submission to Clouds such as Amazon/EC2 (perhaps leveraging the EC2/Nimbus approach), Magellan Cloud (Eucalyptus plug-in).

3. Conclusions

Cloud computing is an emerging technology with virtualization at its core. While Cloud computing offers an attractive model with its service components (PaaS, SaaS, IaaS) and may lack features such as providing a repository service for securely depositing approved VO-specific VMs, adequate network bandwidth for HPC, possibly common interfaces (and an easy way to contextualize a custom image) or strong security model, the virtualization capability alone offers a unique ability, long time ago foreseen as a benefit by experimental groups such as STAR, to "can" their environment in a fully consistent container with operating system, software, middleware, and all services embedded and hence, helping provision remote resources with tested and approved software for Physics result reproducibility and integrity. In our testing of Cloud resources and approaches, we found that Amazon/EC2 (before the appearance of their latest HPC instance support) provides competitive pricing in offering CPU power comparing to University clusters but insufficient data transfer bandwidth to satisfy the demand of HPC. The hybrid approach (Virtualization, Cloud-ification of Grid resources or clusters) provided by Nimbus and the VOC system offers the very attractive feature of allowing to present a standard grid interface to the end-user therefore, allowing for immediate integration of the Cloud paradigm in experimental production workflows with little to no changes. In contrast, condor/VM and Kestrel use a different approach and need to be supplemented with an external job feeder system, but their scalability to larger number of jobs and burden on the local team (contextualization) is heavily reduced by the simplifications. The Kestrel model though has allowed STAR to run stably an operation of a 1,000 jobs scale for a month in a full data mining context (full workflow with runtime gathering of Meta-data as real data production would require). We also observed that the community is rapidly researching and developing interfaces allowing for smooth integration of Cloud and Grid and highlighted a few usage cases of Cloud computing for HPC from partitioning of virtual clusters in the Exascale, easy software provisioning of Tier centers, and an easier distributed computing model for experimental groups focused on delivering stable World-wide renown science for the past decade. We infer Cloud and Grid are not orthogonal approaches and that a merging of both technologies is bound to happen at a fast pace. We predict 10k to 100k Cloud-based production operations would be reachable by the end of 2010. Finally, STAR is already running the most complex simulations and workflows on Cloud, requiring Meta-Data access (database) as well as full event reconstruction, similar to real-data production. In short, and modulo resolving the massive data transfer required for the input, STAR is one step away from being able to run real data production and reconstruction on Cloud in a truly and fully opportunistic manner.

References

[1] The Anatomy of the Grid: Enabling Scalable Virtual Organizations. I. Foster, C. Kesselman, S. Tuecke. *International J. Supercomputer Applications*, 15(3), 2001.
[2] The Open Science Grid consortium, http://www.opensciencegrid.org/.
[3] Knowledge management and virtual organizations. Yogesh Malhorta**,** Idea Group Inc (IGI), 2000, Business & Economics, ISBN 1-878289-73-X.
[4] The STAR collaboration: J. Adams et al., Experimental and theoretical challenges in the search for the quark gluon plasma: The STAR collaboration's critical assessment of the evidence from RHIC collisions. Nuclear Physics A, 757:102, 2005.
[5] The Relativistic Heavy Ion Collider, http://www.bnl.gov/rhic/.
[6] The STAR computing resource plan 2009, STAR Notes CSN0474.
[7] The Particle Data Grid (PPDG) Collaboratory Pilot, http://www.ppdg.net/.
[8] SunGrid and the STAR Experiment, http://www.sun.com/service/sungrid/brookhaven.pdf + Experience with on-demand physics simulations on the Sun Microsystems computing facility (SunGrid) at network.com, J. Lauret et al. 2008. *J. Phys.: Conf. Ser.* 119 052024.
[9] Integrating X-Grid into the HENP distributed computing model, L Hajdu, A Kocoloski, J. Lauret, and M. Miller, 2008. *J. Phys.: Conf. Ser.* 119 072018.

[10] Nimbus and Cloud Computing Meet STAR Production Demands, HPC wire April 02 2009 + Clouds make way for STAR to shine, iSGTW April 8th 2009 + Number Crunching Made Easy, Cloud computing is making high-end computing readily available to researchers in rich and poor nations alike, Newsweek May 2nd 2009 + Computing for the RHIC experiments, CHEP 2009 invited contribution.
[11] STAR Physics—Utilizing the Grid, PPDG report May 2004.
[12] From BNL to KISTI: Establishing High Performance Data Transfer From the U.S. to Asia, RHIC News January 13, 2009.
[13] Setting up a STAR Tier 2 Site at Golias/Prague Farm, P. Chaloupka et al. 2010. *J. Phys.: Conf. Ser.* 219 072031.
[14] rBuildeer.com, a product from rPath Inc., http://www.rpath.com/.
[15] Xen hypervisor, an OpenSource industry standard for virtualization, http://www.xen.org/.
[16] VirtualBox, a general-purpose full virtualizer for x86 hardware, http://www.virtualbox.org/.
[17] National Institute of Standards and Technology, CS division, Cloud computing definitions
[18] Amazon Elastic computing Cloud, Amazon EC2, http://aws.amazon.com/ec2/.
[19] Nimbus, an OpenSource toolkit for enabling IaaS on clusters, http://www.nimbusproject.org/.
[20] M. A. Murphy and S. Goasguen. "Virtual Organization Clusters: Self-Provisioned Clouds on the Grid," submitted to Elsevier Journal of Future Generation Computer Systems special issue on Cloud Computing (in press).
[21] Contextualization in Practice: The Clemson Experience, M. Fenn, S. Goasguen, and J. Lauret, proceedings of 13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (in press).
[22] The Condor project, a high throughput computing scheduler system and its Virtual Machine Applications.
[23] Kestrel: an XMPP-based framework for many task computing applications, L Stour, M. Murphy, and S. Goasgen. In Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers (Portland, Oregon, November 16, 2009). MTAGS '09. ACM, New York, NY, 1–6. doi=http://doi.acm.org/10.1145/1646468.1646479.
[24] MyProxy, an OpenSource software for managing X509 Public Key management Infrastructure (PKI).
[25] Amazon/EC2 instances, http://aws.amazon.com/ec2/instance-types/.
[26] Conceptual Grid Authorization Framework and Classification, Globus Grid Forum report GFD-I.038.
[27] The STAR Meta-Scheduler, http://www.star.bnl.gov/public/comp/Grid/scheduler/.
[28] The Portable Batch System (PBS), Enabling on-demand computing.
[29] The Kernel based Virtual Machine (KVM) system, http://www.linux-kvm.org/page/Main_Page.
[30] OSG Gratia monitoring, http://gratia-osg-prod-reports.opensciencegrid.org/gratia-reporting/.
[31] Gartner's 2009 Hype Cycle Special Report Evaluates Maturity of 1,650 Technologies.
[32] New Amazon EC2 Instance Type—The Cluster Compute Instance, addressing HPC needs.
[33] Magellan Project, Exploring CLOUD Computing for DOE's Scientific Mission.
[34] Computing in the Cloud (CiC), http://www.nsf.gov/pubs/2010/nsf10550/nsf10550.htm.
[35] libcloud, a unified interface to the cloud, http://incubator.apache.org/libcloud/.
[36] Delta.Cloud, many clouds one API, http://deltacloud.org/.
[37] StratusLab, Enhancing Grid Infrastructure with Cloud Computing, http://www.stratuslab.eu/doku.php.
[38] Jabber Technology, http://www.cisco.com/web/about/ac49/ac0/ac1/ac258/JabberInc.html.
[39] ExTENCI: Extending Science Through Enhanced National Cyberinfrastructure.

ADIOS: Powering I/O to extreme scale computing

Q. Liu,[1] S. Klasky,[1] N. Podhorszki,[1] J. Lofstead,[2] H. Abbasi,[2] C.S. Chang,[3] J. Cummings,[4] D. Dinakar,[5] C. Docan,[6] S. Ethier,[7] R. Grout,[8] T. Kordenbrock,[9] Z. Lin,[10] X. Ma,[5] R. Oldfield,[11] M. Parashar,[6] A. Romosan,[13] N. Samatova,[5] K. Schwan,[2] A. Shoshani,[13] Y. Tian,[12] M. Wolf,[2] W. Yu,[12] F. Zhang,[6] F. Zheng[2]

[1]Oak Ridge National Laboratory, TN
[2]Georgia Tech, GA
[3]New York University NY
[4]California Institute of Technology, CA
[5]North Carolina State University, NC
[6]Rutgers University, NJ
[7]Princeton Plasma Physics Laboratory, NJ
[8]National Renewable Energy Laboratory, CO
[9]HP, Nashville, TN
[10]UC Irvine, CA
[11]Sandia National Laboratory, NM
[12]Auburn University, AL
[13]Lawrence Berkeley National Laboratory, CA

Abstract. The Adaptable I/O System (ADIOS) provides a simple, flexible way for scientists to describe the data in their code that may need to be written, read, or processed outside of the running simulation. By providing an XML file, external to the application code, describing the various elements, their types, and how you wish to process them in this run, the routines in the application code can transparently change how they process the data. The goal of this system is to give a level of adaptability such that the scientist can change how the I/O in their code works simply by changing a single entry in the XML file and restarting the code. This brings ADIOS the capability of performing advanced data services such as data staging and adaptive I/O.

1. Introduction

The Adaptable I/O System (ADIOS) [1] developed at Oak Ridge National Laboratory provides a simple, flexible way for scientists to manage their data in the code that may need to be written, read, or processed during simulation runs. ADIOS utilizes an external XML file to describe user data, e.g., data types, sizes and I/O operations. As such, the routines in the user code can transparently change the way they process data. In particular, the XML includes data hierarchy, data type specifications, process grouping and how to process the data. These settings in the XML file are read on code startup and, based upon them, the data will be processed differently. For example, one can select MPI I/O to write out a single shared file for analysis output or POSIX I/O to write out one file per process for checkpoint/restart data. Different setups make the code behave differently without having to either change or recompile the source code. The primary goal of this I/O system is to offer a level of adaptability such that the scientist can change the I/O transport simply by changing a single attribute in the XML file. A high-level illustration of ADIOS is shown in Figure 1 and Figure 2.

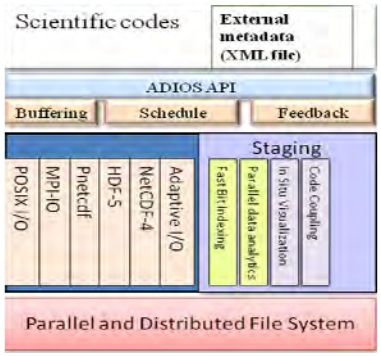


Figure 1. Adaptable I/O System.

```
<adios-config host-language="C">
<adios-group name="temperature" coordination-communicator="comm">
        <var name="NX" type="integer"/>
        <var name="t" type="double" dimensions="NX"/>
</adios-group>
<method group="temperature" method="MPI"/>
<buffer size-MB="1" allocate-time="now"/>
</adios-config>
```

Figure 2. Sample ADIOS XML file.

In ADIOS Each I/O operation is based upon a particular data collection (a.k.a adios group), writing or reading, and then calling close. The following code snippet shows a typical ways of using ADIOS API.

```
adios_open (&group_handle, io_group, "restart.bp", comm)
adios_group_size (group_handle, adios_groupsize, &adios_totalsize);
adios_write (group_handle, "comm", comm)
adios_write (group_handle, "zion", zion)
adios_write (group_handle, "mzeta", mzeta)
adios_close (group_handle)
```

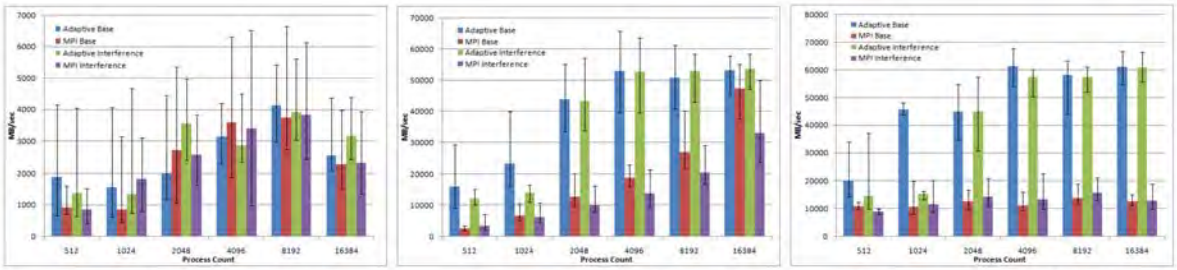
adios_group_size passes per-process size information to the ADIOS library, which is necessary when writing into one shared file by several processes. The second input parameter is the raw data size and the third output parameter is the returned total data size including metadata. The string second parameter to *adios_write* specifies which var in the XML the provided data represents. Note that *adios_close* is considered a "commit" operation. Once it returns, all provided buffers are considered reusable. ADIOS 1.2 and later releases also provide a programmable API, which targets specifically at adaptive-type codes, such as Adaptive Mesh Refinement (AMR) codes. For such applications, the data is usually unknown a priori, for example, the level of refinements. Additional API's are added to allow users to define their data during runtime. Latest I/O methods in ADIOS also offer (1) adaptive I/O (2) data staging, i.e., NSSI staging and DataTap, which are detailed next.

2. Managing I/O Variability in ADIOS

Variability in file system performance due to concurrent use has existed since multi-user operating systems were developed, causing parallel file systems to employ rich caching and other performance management techniques for their internal storage targets. The internal and external interference effects seen in parallel file systems, however, are not adequately addressed by these techniques. We have developed a new set of dynamic and proactive methods for managing I/O interference. These adaptive I/O methods improve I/O performance by dynamically shifting work from heavily used areas of the storage system to those that are more lightly loaded. By using adaptive I/O, we have been able to substantially improve the I/O performance of petascale codes, including that of fusion simulations like GTC, XGC1, GTS, and Pixie3D. These codes generate restart and analysis data every 15 or 30 minutes, on the full scale of the system, and production data size is generally between 64 MB and 200 MB per process. For a typical petascale run of around 150,000 processes, 200 MB per process yields 3 TB to be written every 30 minutes. Staying within a generally acceptable 5% of wall clock time spent in I/O limit, this requires a minimum sustained speed of 35 GB/sec. With the current Lustre limit of a maximum of 160 storage targets for a single file, and a per storage target theoretical maximum performance of around 180 MB/sec, a maximum of only 28 GB/sec can be achieved in

theory, assuming perfectly tuned I/O routines and an otherwise quiet system. Removing this limit can address internal interference, of course, but it does not help with external interference in a busy system. In response, adaptive I/O is designed so as to cope with both internal and external interference effects, the goal being to consistently achieve >50% of peak I/O performance.

Experimental results presented in [2] assess and diagnose the presence and effects of internal and external interference in petascale storage systems. Based on the insights gained from these evaluations, adaptive I/O methods are implemented in the context of the ADIOS I/O middleware, and are now widely deployed for petascale codes. The outcome is a substantial improvement in I/O performance, ranging from around 2x the average performance for a 16384 process run of XGC1 to more than 4.8x for the 16384 process run of Pixie3D with 16 TB output per I/O, all with less variability in the time spent performing I/O (Figure 3).



(a) Small Data (2 MB/process) (b) Large Data (128 MB/process) (c) Extra Large Data (1024 MB/process)

Figure 3. Pixie3D I/O Performance.

3. NSSI Staging in ADIOS

The Network-Scalable Service Interface (NSSI) [5] is a framework for data services developed as part of the the Lightweight File Systems Project [4]. NSSI is a remote procedure call (RPC) based library that is designed to be scalable and handle large I/O. It uses XDR [6] to encode requests and remote direct-memory access (RDMA) to transfer data to and from the client application. The library makes separate RDMA requests for large data segments, which are not XDR encoded. Thus, most copying of request and response data is avoided. The server coordinates the sending and receiving of these large data segments, so it is not overwhelmed by client requests. NSSI currently has implementations for Portals-based networks [3], such as those for the Cray XT series, and InfiniBand networks.

NSSI Staging in ADIOS is composed of two components—a client method and a staging service. The client method differs from other ADIOS methods in that it does not perform any file I/O directly to the file system. Instead, the client method translates ADIOS calls into remote requests to the staging service. Client requests fall into two categories—pass-through and caching. Pass-through requests are synchronous on the staging service and return an error immediately upon failure. Pass-through requests are small, relatively quick operations that will generally become metadata operations for the file system. If the staging service cannot perform these operations, all future operations on this file (including expensive data operations) will fail so the small overhead of these operations is incurred. Caching requests are asynchronous on the staging service. Caching requests could be small or large, but are directly tied to the transfer, placement or transformation of bulk data.

The NSSI Staging service is a parallel application launched concurrently with an ADIOS client application that acts as a proxy for ADIOS procedures. The staging service is itself an ADIOS application, so it can be configured to use any method provided by ADIOS. When the staging service receives a caching request, it pulls the data from the client into a local buffer on the service, then adds the buffer to an aggregation list. When the client indicates that is has finished writing data, the staging service performs as much aggregation as possible, then executes the necessary ADIOS calls to write the data to persistent storage. The idea is to provide a large enough buffer in the network to absorb

large bursts of I/O, then filter the I/O to the storage system during the compute-intensive phase of the application. With sufficient memory in the network, the application should observe effective I/O rates at the speed of the network instead of the storage system.

Staging performance has been measured using an ADIOS micro-benchmark that generates a single 3D global array on the Jaguar Cray XT5 system at the Oak Ridge National Laboratory. Figure 4 shows the measured performance results of two different experiments: writing a single shared file using the NetCDF4 method (no staging) and writing a single shared file using NetCDF4 through the staging service. Each experiment wrote 10MB per compute core to the file. The staging service required 1/32 additional compute nodes to make sure the data remained in-core for the output dump. The Lustre file system was configured with a stripe size of 4MB and a stripe count of 128. Results on Jaguar show that unstaged NetCDF4 to a single shared file has miserable performance, maxing out at 251MB/s. The staged NetCDF4 was able to achieve an "effective" 8.05GB/s to a single shared file. This is the rate observed by the application as the time to transfer the data from the application to the set of staging service nodes. In cases where the staging service has sufficient memory and a sufficiently long compute phase of the application, the staging service results in a 60x improvement in I/O performance with only 1/32 additional compute resources.

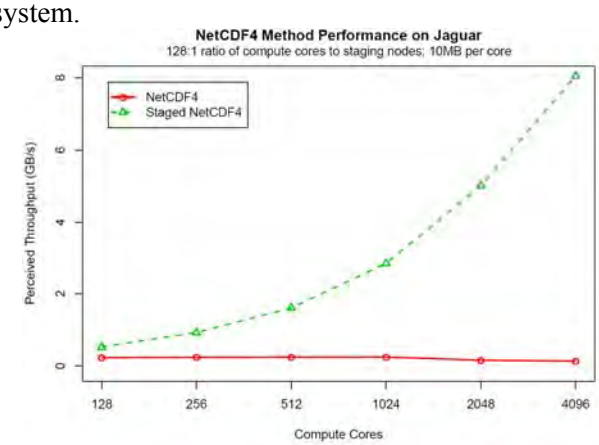


Figure 4. By staging ADIOS data in a partition of compute nodes, the application sees effective I/O rates between 2x and 60x larger than writing directly to the parallel file system.

4. Data Staging with EnStage

EnStage is an enhanced data staging infrastructure developed at the Oak Ridge National Laboratory and Georgia Tech. The motivations underlying the research into EnStage are the increasingly stringent demands of I/O from current and next generation scientific applications. With very high data rates the storage subsystem becomes a bottleneck limiting the performance and scalability of the entire application. Alleviating this bottleneck can be accomplished by either improving the storage subsystem by adding faster disks, more OST (Object Storage Targets), or the platform can present a different mechanism for I/O that can utilize existing resources to improve the I/O situation.

4.1. Staging

The later approach has led us to consider the advantages of using staging for I/O. We define staging as the utilization of additional compute area resources for providing transient storage and compute capacity. Thus instead of the application "writing" data to storage, staging provides a level of indirection to the process. In the staging approach to I/O the application outputs the data to staging, where it can undergo further processing and the staging area is responsible for the final transfer of the data to the storage subsystem [8]. For the purpose of clarity, we next define some of the terms for the reader.

Staging Area: A collection of nodes that are selected to serve as a transient location for temporary storage and processing. These nodes are not functionally different from other nodes used in the computational application, but are re-purposed to enhance the I/O pipeline.

DataTap: The DataTap is a lightweight application library that enables the use of the "DataTap" ADIOS transport method for the application. DataTap utilizes RDMA to enable a request-read data

transport model, allowing for the application to scale without placing restriction on the size of the staging area.

DataStager: The DataStager [7] is the counterpart of the DataTap residing within the staging area. The DataStager is responsible for reading the data from the DataTap, as well as aggregating the data for further processing within the staging area.

4.2. DataTap and DataStager

The DataTap is the foundation of the staging transport within ADIOS. Instead of transferring the entire output buffer while the application waits for completion, the DataTap transport only transmits a small request message to the staging area. Once received, this message is queued up in the staging area and eventually serviced by the DataStager. The DataStager processes each of the queued requests, utilizing a variety of selection criteria to pick the more appropriate request for servicing. Once a request has been selected the DataStager issues a remote read request to the source. The data transfer is thus completed and the DataStager passes the message to a user defined handler.

A significant portion of the DataStager is devoted to selecting the right request from the queue. The first step in this determination is a check to see if the available memory in the DataStager can buffer the entire data message. As data is read in, is processed and finally retired, the level of available memory in the system varies greatly. Due to the large difference in the scale of the memory available to the application and to the memory available to the staging area, moving the control of data movement to the staging area is the only efficient method for handling this imbalance. While the most basic memory check will apply to all requests there are a number of checks that the user can enable. DataStager issues concurrent data read requests when sufficient local memory is available, but this approach can result in unnecessary perturbation with intra-application communication. Another, more complex, check is for the current phase of the source node. Scientific applications operate in phases and there are periods of time when the node is only involved in computation. Transferring the data during this time avoids the problems associated with interference on a shared network link. This perturbation avoidance scheduler allows the staging area to be used with very scalable applications without a significant performance penalty from interference.

Acknowledgement

This work is jointly supported by US DOE Office of Fusion Energy Science and Office of Advanced Scientific Computing Research.

References

[1] Lofstead, J, Zheng, F, Klasky, S, and Schwan, K. “Adaptable, Metadata Rich IO Methods for Portable High Performance IO”, In *Proceedings of IPDPS*, Rome, Italy, May, 2009.

[2] J. Lofstead, F. Zheng, Q. Liu, S. Klasky, R. Oldfield, T. Kordenbrock, K. Schwan, and M. Wolf, “Managing Variability in the IO Performance of Petascale Storage Systems,” to appear in SC 2010.

[3] Ron Brightwell, Tramm Hudson, Arthur B. Maccabe, and Rolf Riesen, The Portals 3.0 message passing interface. Technical Report SAND99-2959, Sandia National Laboratories, November 1999.

[4] Ron A. Oldfield, Arthur B. Maccabe, Sarala Arunagiri, Todd Kordenbrock, Rolf Riesen, Lee Ward, and Patrick Widener. Lightweight I/O for scientific applications. In *Proceedings of the IEEE International Conference on Cluster Computing*, Barcelona, Spain, September 2006.

[5] Ron A. Oldfield, et al., Efficient data-movement for lightweight I/O. In *Proceedings of the 2006 High Performance I/O Techniques and Deployment of Very Large Scale I/O Systems*, Barcelona, Spain, September 2006.

[6] Sun Microsystems, Inc. RFC 1014: XDR: External Data Representation standard, June 1987.

[7] Abbasi, H., Wolf, M., Eisenhauer, G., Klasky, S., Schwan, K., and Zheng, F. 2009. DataStager: scalable data staging services for petascale applications. In *Proceedings of HPDC*, Germany, June 2009.

[8] H. Abbasi, J. Lofstead, F. Zheng, S. Klasky, K. Schwan, and M. Wolf, "Extending I/O through High Performance Data Services." *Cluster Computing 2009*, New Orleans, LA. August 2009.

Policy-based data management

Reagan Moore,[1] Arcot Rajasekar,[1] Mike Wan,[2] Wayne Schroeder,[2] Chien-Yi Hou[1]

Data Intensive Cyber Environments Center
[1]University of North Carolina at Chapel Hill
[2]University of California at San Diego

Abstract. Next generation data grid technologies automate policy enforcement, as exemplified by the integrated Rule-Oriented Data System (iRODS). The combination of data grid virtualization mechanisms with policy based data management makes it possible to automate administrative functions, enforce policies such as distribution, retention, disposition, replication, and periodically validate assessment criteria. In this paper, we provide brief outlines of production-level use cases for iRODS being undertaken around the world.

1. Introduction

The **integrated Rule-Oriented Data System** (iRODS) [1–2] is data grid middleware. It provides mechanisms for managing, querying, accessing, and preserving data that are distributed across multiple storage resources. The iRODS system applies policy-based controls when performing these functions. Specifically, the iRODS system provides the following capabilities:

- Global persistent identifiers for naming objects.
- Support for metadata to identify system-level physical properties of the stored data.
- Support for descriptive metadata to enable discovery through simple query mechanisms.
- Standard access mechanisms such as browsers, shell commands, load libraries, Java, C library calls, file system interfaces, digital libraries, workflows, and web services.
- Storage repository abstraction for storing files on tape, disk, and cloud storage.
- Inter-realm authentication system for secure access to remote storage systems.
- Support for replication and synchronization of files between resource sites.
- Support for caching copies of files onto a local storage system.
- Support for aggregating files to optimize management of large numbers of small files.
- Access controls and audit trails to control and track data usage.
- Support for execution of remote operations for data sub-setting, metadata extraction, indexing, remote data movement, etc. using workflows composed from micro-services.
- Support for diverse I/O models for files.
- Support for federation of data grids.

The iRODS data grid system consists of four types of peer-to-peer servers. A metadata catalog server, called the iCAT server, provides the metadata and abstraction services. Resource servers provide access to one or more storage resources through a built-in distributed rule engine. The rules can be viewed as defining processing pipelines or workflows. The building blocks for the iRODS rules are micro-services—small, well-defined procedures or functions that perform a certain task. Extensibility and customizability are achieved by encoding server-side operations (including the main

access APIs) into sequences of micro-services. The micro-services are controlled by user-defined and/or administrator-defined Event-Condition-Action rules similar to those found in active databases. For example, one may encode a rule that when accessing a data object from a collection, additional authorization checks need to be made. These authorization checks can be encoded as a set of micro-services with triggers that fire based on current operating conditions. In this way, one can control access to sensitive data based on rules and can escalate or reduce authorization levels dynamically as the situation warrants. The iRODS rule engine design builds upon the application of theories and concepts from a wide range of well-known paradigms from fields such as active databases, transactional systems, logic programming, business rule systems, constraint-management systems, workflows, service-oriented architecture and program verification. Apart from iRES servers and an iCAT server, iRODS also has two other servers: iSEC for scheduling and executing queued rules, and iXMS for providing a message-passing framework between micro-services. Figure 1 shows the various components of the iRODS system as well as some of its user interfaces.

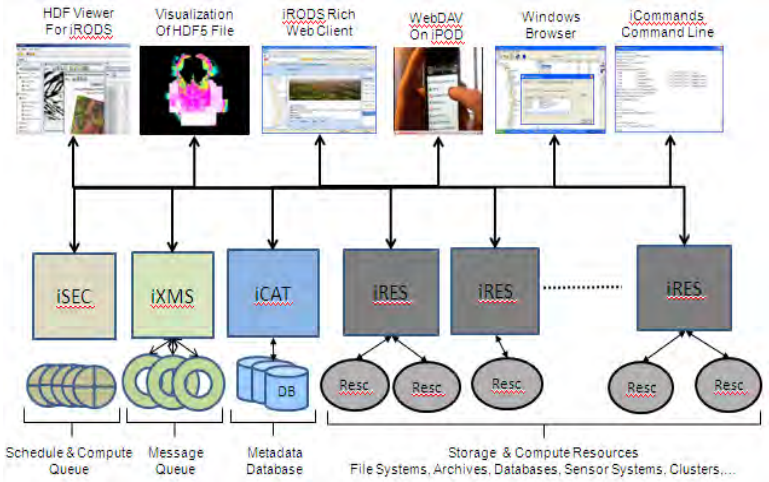


Figure 1. iRODS System Architecture.

The iRODS system is in production use in multiple projects including the US National Archives Transcontinental Persistent Archive Prototype (TPAP) [3], the NSF Science of Learning Centers [4], the Australian Research Collaboration Services [**Error! Reference source not found.**], and the SHAMAN project in UK [6]. More than 2 PetaBytes are managed by iRODS and its forerunner the Storage Resource Broker at the IN2P3 supercomputing center in Nice France [7]. Several large-scale projects such as the iPlant Collaborative [8], the Ocean Observatories Initiative [9], and the Consortium of Universities for the Advancement of Hydrologic Science (CUAHSI) [10] are starting to use the iRODS system for their production data sharing and archiving environment.

2. Use Cases from Ongoing Large-Scale Production Environments

One can categorize the usage models for iRODS into different types:

a) Uniform name space for a large wide-area file system for use by users who perform analyses from multiple computing resources,
b) Sharing of files among cooperating small discipline-centric groups,
c) Large-scale data sharing across large groups of users, who share data from multiple sensors or share data from computer simulations,
d) Publication of data in a digital library environment with curation, organization, and standardized metadata capabilities, and
e) Long-term storage and access to data in a preservation environment.

The use cases that we describe are from existing production systems run by several organizations that are not part of the iRODS DICE group. The feedback from these users provides us with a set of assurances about the usability of the iRODS data grid middleware, and indeed provides a strong indication of its extensibility and ability to span requirements across multiple disciplines.

2.1. PetaShare

(*Uniform Name Space*) PetaShare [11] is an NSF funded large-scale data management project with the goal of making data available for multi-disciplinary scientists across the state of Louisiana for data-intensive analysis and visualization. The project provides data cyber infrastructure to transparently handle all low-level details of file and data management across multiple sites and provide transparent data sharing, retrieval and archival mechanisms to promote scientific computing. Petashare uses an enhanced version of iRODS to provide a global name space and efficient data access among geographically distributed storage resources. PetaShare applies the Stork [12] data placement scheduler that takes the responsibility of managing data resources and scheduling data tasks from the user and performs these tasks transparently. For the users, PetaShare provides light-weight interfaces called PetaFs (Fuse-based file interface), Petashell (based on Parrot as a unix-type shell), and Pcommands (based on iRODS iCommands). The PetaFs and Petashell interfaces provide Unix-type standardized access as if the data were located on local disk. The Pcommands provide a richer metadata-based discovery and access mechanism.

PetaShare is currently deployed across five state universities and two health sciences centers in Louisiana. These institutions include Louisiana State University (LSU), Tulane University, University of New Orleans, University of Louisiana at Lafayette, Louisiana Tech University, and LSU Health Sciences Centers in New Orleans and Shreveport. PetaShare manages approximately 300 Terabytes of disk storage distributed across the PetaShare network sites as well as 400 Terabytes of tape storage centrally located at the LSU campus. PetaShare uses the LONI communication network, a statewide 40 Gbps fiber-optic network in Louisiana. The PetaShare project currently supports more than 25 multi-disciplinary projects from coastal hazard prediction to DNA sequencing to computational fluid dynamics.

2.2. CC-IN2P3 Data Grids

(*Small-Scale Data Sharing, Large-Scale Data Sharing*) CC-IN2P3 [13] is a French national supercomputing center located at Lyon. It provides computing and storage facilities for French and international scientists in the fields of high energy physics, nuclear physics, astrophysics, biology and biomedical applications as well as in arts and humanities. iRODS is being used in production at CC-IN2P3 on multiple projects including:

a) TIDRA (Traitement Informatique Distribué en Rhône-Alpes) provides support for five major laboratories located in the Rhône-Alpes area, and uses iRODS to serve biology applications (phylogeny), biomedical imaging applications (mice) and human data (heart and lung studies). The growing set of applications is expected to reach 20 TB by 2010.
b) Adonis is a nationally funded project providing computing and storage facilities for Arts and Humanities and is linked to other European projects such as DARIA. iRODS provides archival functionality, data access through the web, access to data across batch processing farms for riverbed studies in geography, movie simulations of ancient monuments, etc. The iRODS rule-based capability is used for automating management, distribution and extraction processes. With currently more than 20 TB, the Adonis project expects to grow to 100 TB by 2010.

The iRODS service at CC-IN2P3 is supported by 10 servers, two used for running iCAT services, 6 for storage services and two for running the Oracle databases needed by iCAT. To mitigate single-point failure and provide redundancy, a DNS load-balancing alias (developed originally at SLAC) is used. The two iCAT servers in this DNS alias provide redundancy and fault-tolerance. An iRODS Health monitoring system developed at IN2P3 (and shared with the iRODS community) is used to check the performance and working of the various components of the distributed iRODS data grid. Several other projects are also supported by CC-IN2P3 and expect to host more than 300TB of data by 2010. CC-IN2P3 also expects to migrate its 2 Petabytes of data holdings (from BaBar and Lattice QCD) on the Storage Resource Broker (SRB) into iRODS with expected completion of this migration in 2012.

2.3. Dataverse Archive

(*Long-Term Preservation*) The archivists at the H. W. Odum Institute for Research in Social Science at the University of North Carolina at Chapel Hill, use an open-source web publishing platform called Dataverse [14] to publish their extensive collection of files related to social science research studies. As a component of their preservation strategy, Odum has developed an inter-operability mechanism between Dataverse and iRODS that copies the contents of an entire Dataverse archive into iRODS. The transfer process also extracts descriptive metadata and stores them in the iCAT metadata catalog. The transfer of contents between Dataverse and iRODS is accomplished by using methods based on OAI-PMH and HTTP protocols. An additional advantage of this transfer is that the users now have the rich set of interfaces of iRODS to access data including keyword-based search and discovery on the serialized copy of the Dataverse archive. Moreover, the integration allowed the Dataverse administrators to enforce data preservation policies across their holdings including redundancy, integrity checking, technology migration, etc.

2.4. The ARCS Data Fabric

(*Large-Scale Data Sharing*) The Australian Research Collaboration Service (ARCS) Data Fabric [15] is national data infrastructure deployed in Australia for researchers to easily store their research data, and share them across institutional boundaries. ARCS Data Fabric is conceived as a generic service crossing multiple research disciplines that is available to every Australian researcher and their international collaborators. ARCS supports two types of access (both built with iRODS as an underlying infrastructure): webDrive is a two-layered access interface, whose back-end is iRODS which brokers storage at distributed member sites and whose front-end is a WebDAV server-browser system which allows transparent access to and ingestion of files across the ARCS Data Fabric. Apart from uploading and downloading functionalities, the WebDAV interface also provides access control mechanisms, metadata ingestion and discovery for files and collections, as well as a 'trash can' feature. The authentication mechanism of the ARCS webDrive supports the Australian Access Federation (AAF) standardized methods and technologies. Apart from the webDrive interface, ARCS also provides an access mechanism through the OPeNDAP protocol based on THREDDS Data Server. IRODS is used at the back-end to provide data access. Currently, the ARCS Data Fabric runs a seven-node iRODS zone with nodes set up at each of the capital cities in Australia. Currently more than 18TB of data is registered into the ARCS DF and is used by more than 280 users.

2.5. TELDAP

(*Long-Term Preservation*) Taiwan e-Learning and Digital Archives Program (TELDAP) [16] collects and integrates digital material from archives from different disciplinary institutes, museums and research groups from across Taiwan. As part of a Digital Archives Remote-Backup (DARB) project, a Sinica Data preservation System (SIDPS) is being deployed across Taiwan to capture and store digital archives and content from diverse fields including history, biology, ethnology, education and language. The SIDPS system uses the iRODS as a backbone for providing rule-based policy mechanisms, optimized data transfer functions, and to flexibly coordinate data stored within distributed environments in different institutes across Taiwan. To provide a better user experience for its users, the DARB project has developed a specialized interface called UrSpace that provides a GUI data management tool for data manipulation, data transfer, encryption, search and metadata management. The DARB also supports a synchronization mechanism that can be used for synchronizing files from user spaces with iRODS. These tools provide international language support so that files and metadata with Chinese characters will be handled correctly. The DARB project also uses a monitoring system called SIMS for detecting problems in the iRODS systems in order to maintain high production quality.

3. Conclusion

The NSF funded iRODS system has been under deployment for more than two years and has been adopted by several large-scale users. We have briefly described a few of these usage models that support different data sharing models, from name space management to data sharing to preservation and digital library maintenance. The extensibility and modularity of iRODS makes it a sustainable software model powered with policy-guided data management using micro-service oriented architecture. More information on iRODS can be found at www.irods.org

Acknowledgement

The research results in this paper were funded by the NSF Office of Cyberinfrastructure OCI-0848296 grant, "NARA Transcontinental Persistent Archive Prototype." The iRODS technology development has been funded by NSF SDCI 0721400, "SDCI Data Improvement: Data Grids for Community Driven Applications."

References

[1] Integrated Rule Oriented Data Systems, https://www.irods.org/index.php.
[2] Rajasekar A., R. Moore, C-Y. Hou, L. Christopher, R. Marciano, A. de Torcy, M. Wan, W. Schroeder, S-Y. Chen, L. Gilbert, P. Tooby, and B. Zhu, "iRODS Primer: integrated Rule-Oriented Data Systems," ISBN: 9781608453337, Book published by Morgan-Claypool Publishers, San Rafael, CA., January 2010.
[3] NARA Transcontinental Persistent Archive Prototype, http://www.dlib.org/dlib/july07/07inbrief.html.
[4] NSF Science of Learning Centers, http://silccenter.org/nsf_slcs_index.html.
[5] S. Zhang, P. Coddington, and A. Wendelborn, "Davis: A Generic Interface for SRB and iRODS," www.dhpc.adelaide.edu.au/reports/197/dhpc-197.pdf.
[6] EU SHAMAN Project, http://shaman-ip.eu/shaman.
[7] J-Y Nief, "SRB fête ses 2 péta-octets au CC-IN2P3," in IN2P3 Information Letters, No 9. January 2010, http://informatique.in2p3.fr/li/spip.php?article43.
[8] NSF iPlant Collaborative, http://www.iplantcollaborative.org/.
[9] NSF Oceans Observatories Initiative, http://www.whoi.edu/programs/ooi/.
[10] NSF CUAHSI Project, http://www.cuahsi.org/.
[11] Petashare, http://www.petashare.org/.
[12] Stork, http://stork.cct.lsu.edu/index.php.
[13] IN2P3, Institut National de Physique Nucleaire et de Physique des Particules, http://www.in2p3.fr/#.
[14] Dataverse, http://thedata.org/.
[15] ARCS Australian Research Collaboration Service, http://www.arcs.org.au/.
[16] TELDAP, http://teldap.tw/en/.

Scientific Data Management: Challenges and Approaches in the Extreme Scale Era

Arie Shoshani (LBNL), Scott Klasky (ORNL), Rob Ross (ANL)

Abstract. Exascale computing will prove to be challenging for scientific data management, given the expected increased volume of data from simulations, and the current expectation of the extreme scale computer architecture and storage systems. Scientific data management includes all of the associated problems of managing the data from its inception, to how it moves from the data generator (exascale resource) to the storage system, along with the concurrent analysis, reduction, indexing, and provenance collection, during and after the data is generated. An associated challenge is the reduction of the energy consumption of the entire I/O pipeline, and the integration of these techniques into a complete end-to-end scientific data management system. In this paper we explore techniques that can create, manage, and reduce energy to help mitigate these challenges.

We begin by explaining what makes the problem of managing scientific data unique. We follow that by delineating the scientific data challenges at extreme scale. Then we provide many examples of technologies developed by the Scientific Data Management (SDM) Center since its inception that could be applied or modified for extreme scale challenges, and implications from SDM center experience. We then conclude with a summary of ideas for handling the coming flood of data.

1. Introduction

1.1. What is Scientific Data Management?

Managing scientific data has been identified by the scientific community as one of the most important emerging needs because of the sheer volume and increasing complexity of data being collected. Effectively generating, managing, and analyzing this information requires a comprehensive, end-to-end approach to data management that encompasses all of the stages from the initial data acquisition to the final analysis of the data. Fortunately, the data management problems encountered by most scientific domains are common enough to be addressed through shared technology solutions. Based on community input, we have identified three significant requirements. First, provide more efficient access to storage systems. Second, scientists require technologies to facilitate better understanding of their data, in particular the ability to effectively perform complex data analysis and searches over extremely large data sets. Finally, scientists require tools to collect and store results, automating the process of provenance storage, data-processing, and data analysis. Managing, storing, searching, and keeping track of these large datasets requires specialized techniques. Specifically, the following four aspects make the problem addressed by Scientific Data Management unique.

(1) Representing the scientific data model. Many scientific domains represent data in space and time. The majority of the data that is created from large scale computations can be broken into two categories: checkpoint-restart data, and analysis/visualization data. Typically the output data from these simulations is written through either: POSIX, MPI-IO, HDF5, netCDF, or similar formats. Data can then be organized in a data model which can often be represented as regular or irregular meshes

for simulation data. The organization of this data under a data model provides more information that storage software can use to to efficiently manage the underlying data.

(2) Managing I/O for large-scale simulations. Many simulation codes produce large volumes of data on large-scale parallel machines, such as leadership class machines. Already today, on machines with 100,000 cores, the volume is measured in hundreds of TB's per day. Dumping such volume of data to a parallel file system is often a bottleneck, where the computation on the nodes is waiting for the I/O to complete. Various techniques have been developed to help with this problem, such as coordinating I/O from multiple nodes (collective I/O), performing the I/O concurrently with the computation (asynchronous I/O), and lining up the data on regular boundaries to avoid lock contention on the parallel file system.

(3) Accelerating efficiency of access. Simulation data is typically generated one time step at a time over a large number of processors. For example, climate models generate tens of variables (temperature, pressure, etc.) over the entire globe for a time step, and then proceed to the next. However, when the data is accessed and analyzed variables are often needed one at a time over all the time steps. Maximizing the efficiency of the analysis tasks requires the data to be reorganized into data structures that are different from the way the data was generated. Scientists typically wish to search this data over several variables, in order to see correlations which provide scientific insight. This requires efficient indexing of the data over each variable to facilitate near real time searches over combination of variables in order to identify regions of interest, such as finding turbulent regions, flame fronts, etc.

(4) Facilitating scientific data analysis. While much of the attention is given to running on extreme scale architectures to model the applications of interest, it is necessary to consider the data analysis as an integral part of the scientific discovery process. The scientific data analysis process involves multiple techniquess, including data exploration algorithms (such as dimensionality reduction, decision trees), data mining (such as graph algorithms, clustering, classification, anomaly detection), data management algorithms (such as indexing, data transformation, data compression), and visualization (such as graphs, contour plots, parallel-coordinate plots, 3D rotation and movies). Furthermore, the data analysis process is often a cyclical activity, which requires tracking the steps used in the analysis and automating the process using technologies such as workflow management. The techniques for streamlining and accelerating the analysis of scientific data are part of the scientific data management challenge.

1.2. Current practice in managing scientific data

Simulations have been referred to as "the third pillar of science," in addition to theory and experiments. Simulations involve the development of mathematical models that are intended to represent the scientific phenomena or problems investigated, and use numerical methods to run such models on large supercomputers to achieve sufficient granularity to achieve accurate modeling. If observational/experimental data is available, then the simulation results need to be validated against them as well.

The typical process of investigating scientific problem includes the following steps, and shown schematically in Fig. 1:

- Run large-scale simulations on large supercomputers
- Dump data on parallel disk systems
- Export data to archives
- Move data to users' sites – usually selected subsets
- Perform data manipulations and analysis on mid-size clusters
- Collect experimental / observational data
- Move to analysis sites
- Perform comparison of experimental/observational to validate simulation data

Note that the site where the data is collected is not necessarily the same site where the analysis is performed, and that the site where observational/experimental data are collected is usually a different site as well. That implies a great deal of data movement over the wide-area network when large simulations are performed. At current rates of Terabytes per simulation run, this is still a manageable task, and specialized infrastructures, such as the Earth System Grid (ESG), have been setup to manage the discovery and distribution of simulation and observational data around the globe. However, when data volumes per simulation run are expected to reach multiple Petabytes, this approach is no longer practical. New techniques are needed to address this extreme scale challenge.

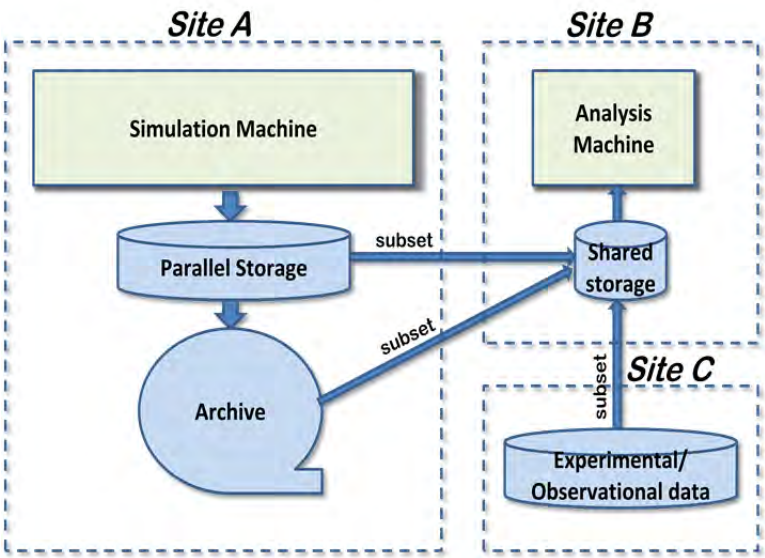


Figure 1. Current practice of the scientific exploration process

2. The scientific data extreme scale challenges

2.1. The I/O challenge

An unavoidable issue in HPC systems is the disparity between the rate at which we can calculate results and the rate at which we can store data persistently. At the root of this issue is the fact that the rate at which storage devices (i.e., disk drives) improve in performance is much lower than the rate at which HPC systems improve in terms of FLOPS, as shown in Fig. 2. This disparity means that increasing the number of drives must be used in each successive deployment to match improvements on the compute side. New persistent memory technologies (e.g., NAND Flash, Phase Change Memory) promise easier construction of high bandwidth aggregate storage units, but their cost per byte of storage is not predicted to drop enough in the near future to replace disk drives.

The I/O challenge, then, is to provide enough on-line storage for application datasets and ensure that on-line storage is accessible for reading and writing at rates that do not leave the computer idle for extended periods. At the extreme scale, the expectation is that on-line storage will be on the order of 500-1000 Petabytes accessible at a peak rate of 60 Terabytes per second. Meeting this challenge will require novel hardware and software architectures. On the hardware side, persistent memory must be placed strategically in the system to serve as a short-term data sink for information flowing off the system, allowing computation to rapidly resume. Meanwhile, new storage software will asynchronously stage data off the system. This software must interact with the operating system and networking layers to ensure that noise is not introduced into the computation, and it must be supported by a storage system that facilitates unprecedented concurrency in I/O.

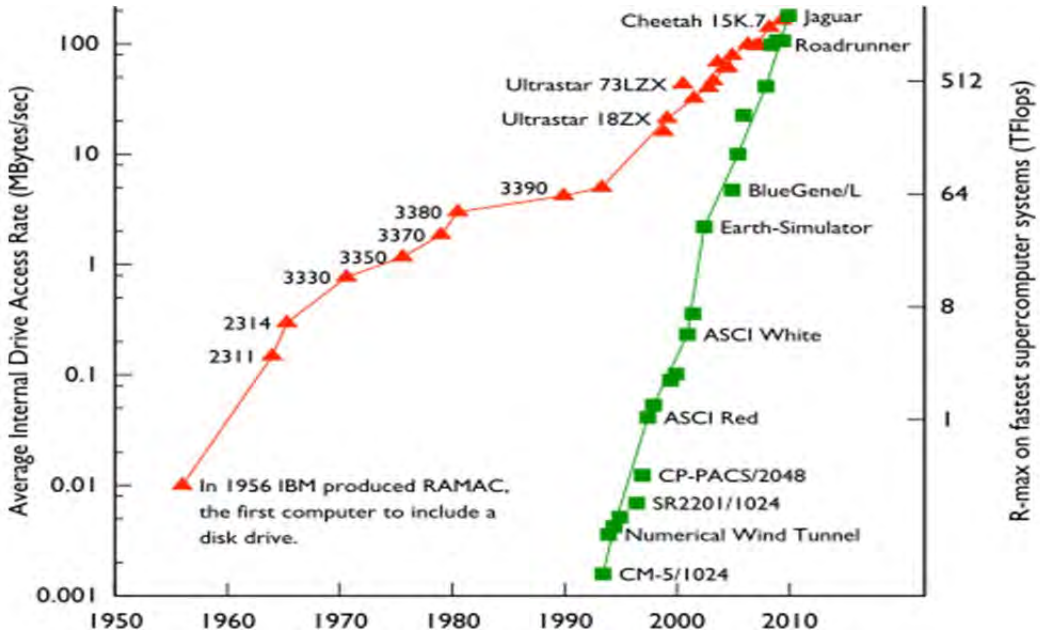


Figure 2. The rate of performance of disks is much lower than compute rates (Thanks to R. Freitas of IBM Almaden Research Center for providing some of the data)

2.2. The data analysis challenge

The volume of data generated by simulations is proportional to: (1) the FLOPS of the HPC system, (2) the memory on the system, and (3) the underlying computational model used in the code. In the extreme scale, it is prohibitive to follow the traditional methods of getting the data out to external storage comprised of disks and/or tertiary storage, and analyze the data later. Instead, two complementary approaches of data analysis need to be considered: (1) "in-situ analysis" - identifying what parts of the analysis can be done in-situ, i.e., while the data is generated; and (2) "preparation for off-line analysis" - what can be done to prepare the data (reduce, summarize, transpose, index, etc.) for off-line processing after the data is stored on external storage. We term the combination of (1) and (2) as an I/O pipeline.

Data analysis, in general, falls into two categories that fit the above approaches. The first approach of "in-situ analysis" can be used for performing many analysis tasks for ***finding known or expected patterns***. The second approach of "preparation for off-line analysis" is in order to facilitate iterative exploratory analysis processes of ***looking for unknown patterns or features*** in the data. These approaches can often benefit from common tasks, such as summarizing the data in-situ for monitoring progress of the simulation in near real-time, as well as using such summaries as a data reduction methods for further off-line analysis.

As the size of simulation, observational, and experimental datasets grows in the extreme scale, many of the existing technologies are not practical for both on-line and off-line data analysis and knowledge discovery processes. Solutions need to take advantage of data reduction methods, parallel processing, smart navigation, summarization, and manipulations of the massive datasets. New methods for achieving better efficiency of searching, processing, exploring, and displaying information are needed.

2.3. The energy reduction challenge

Current focus in the design of future extreme scale machines is on using low power chips. However, even with such low power chips there is a high consumption of energy for **data movement**. According

to recent publications (see for example [7]), about 70% of the extreme scale system power consumption will be due to data movement. Data movement includes three categories: within-node data movement, inter-node data movement, and memory-to-disk data movement. Within-node data movement refers to using the 5 levels of memory: registers, L1, L2, L3 cache, and main memory. Moving data to main memory is 200X more expensive in power consumption than moving it to L1 only. This suggests the need to control where to save data on a temporary basis when performing various in-situ analysis and summarization functions. That will require the development of programming languages that provide explicit references to the levels of memory within-node.

While within-node to main memory requires about 10 cycles, inter-node data movement is in the order of 100's of cycles. This ratio of 1:10 or higher suggests the need to minimize such data movement. This requires the development of algorithms that minimize data movement for various in-situ analysis tasks, as well as summarization, transposition, and indexing.

Data movement to hard drives (disk) requires about 10,000 cycles, and the corresponding energy consumption. Obviously, minimizing what is stored on disk is a useful technique, but in addition it is important to avoid reading back from disk unnecessarily. For example, rather than dumping data to disk, and then generating indexes (which requires reading the entire dataset), it is better to generate the index in-situ before the data in written out to disk. Such indexes also help minimize the data read from disk for subsequent exploratory analysis.

3. Overview of successful technologies in the SDM center

Our approach to scientific data management is to employ an evolutionary development and deployment process: from research through prototypes to deployment and infrastructure. Accordingly, we have organized our activities in three layers that abstract the end-to-end data flow described above. We labeled the layers (from bottom to top in Fig. 3):

- Storage Efficient Access (SEA)
- Data Mining and Analysis (DMA)
- Scientific Process Automation (SPA)

The SEA layer is immediately on top of hardware, operating systems, file systems, and mass storage systems, and provides parallel data access technology, and transparent access to archival storage. The DMA layer, which builds on the functionality of the SEA layer, consists of indexing, feature identification, and parallel statistical analysis technology. The SPA layer, which is on top of the DMA layer, provides the ability to compose scientific workflows from the components in the DMA layer as well as application specific modules.

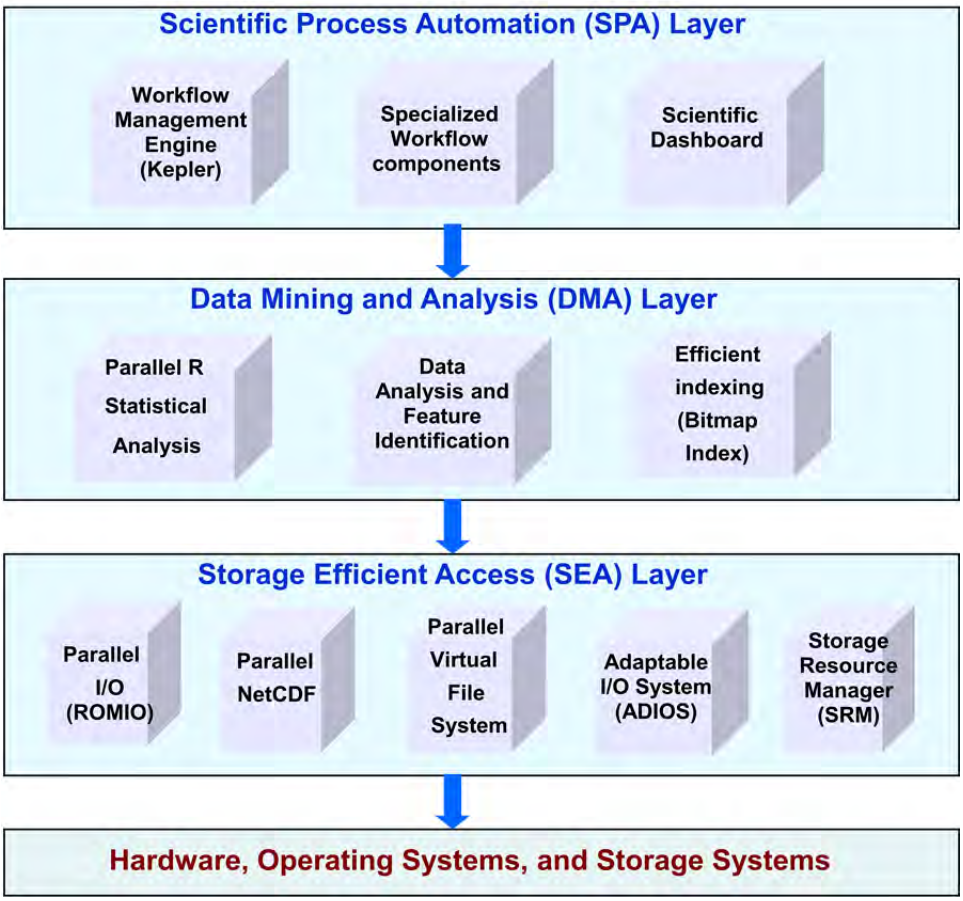


Figure 3. The three layers of technologies in the SDM center and the technologies supported (shown in the labeled boxes).

4. Implications from SDM center experience

4.1. Asynchronous I/O and Combining I/O Operations

While great benefits have been seen for applications utilizing collective I/O routines, it is not always convenient, or even possible, for applications to adopt these interfaces. To provide most of the benefits of the techniques underlying collective I/O implementations to the widest array of applications, new software that combines and asynchronously stages data is necessary. This software is referred to as I/O delegation or I/O staging software.

One such implementation developed as part of the SDM center, from Nisar et.al. [14], is a software layer in MPI-IO that utilizes a small subset of compute nodes (I/O delegates) as a location for data combining and to drive I/O. This layer is implemented at the bottom layer of the ROMIO MPI-IO implementation, where it intercepts all the system I/O requests initiated by ROMIO and redirects them to delegate nodes. The researchers incorporated a static approach to mapping file regions to I/O delegates that was proposed in our earlier work on MPI-IO file domains [9]. Experiments using the FLASH and S3D I/O kernels were performed on two parallel machines: Abe at NCSA and Franklin at NERSC. Their experiments show that using MPI independent I/O functions in the two application kernels can even outperform the same kernels using collective I/O, with the observed improvement ranging from 2.5 times to 15 times better I/O bandwidth.

Beyond the allocation of compute nodes and integration of I/O delegation and caching into I/O middleware like MPI-IO, techniques in the context of I/O forwarding have also been investigated. I/O forwarding software, such as the CNK on IBM Blue Gene/P systems and DVS on Cray XT systems, provides a conduit for transferring I/O operations on clients out to gateway, or I/O nodes that have direct access to storage systems. As part of a DOE/NSF funded collaborative effort a portable I/O forwarding system called IOFSL (I/O Forwarding Software Layer) is being developed in order to investigate how techniques that have proven successful in I/O delegation work can be applied in this new context [15, 21].

4.2. Taking advantage of I/O libraries for in-situ processing

Efficient Scientific Data Management (SDM) is both a necessity and a key enabler for conducting science at scale. This is because SDM goes beyond simply performing input and output for large-scale simulations. It also helps scientists create, manage, and use the entire data output pipeline, from the time data is generated by high end simulations, to the time it is analyzed and visualized. When doing so, efficiency is a necessity, because the large-scale simulations envisioned to run on future extreme scale machines will generate immense amounts of output data. At the same time, however, SDM is a key enabler for science because it is precisely these outputs that hold the key to future scientific discovery, and those outputs only become useful for attaining scientific goals and supporting scientific inquiry after they have been analyzed, evaluated, understood, visualized, stored, and/or transformed into the inputs required by coupled simulations.

The critical importance to scientific discovery of the output data generated by future extreme scale simulations has prompted us to develop a componentized approach to data management. This approach, embodied in the Adaptable I/O System (ADIOS), not only enables high performance I/O at scale, but it also constitutes the beginning of a scalable approach to constructing, managing, and using the entire output pipelines used by high performance simulations. The ADIOS componentized output enables portable high performance across multiple machines, using diverse transports, and supporting both custom and standard file formats and data representations [12]. Furthermore, the PreDatA framework for immediate data analysis [23] interfacing with ADIOS permits users to efficiently apply to output, as it is being generated – in-transit – data annotation, indexing, reorganization, formatting, and immediate analysis actions, using a streaming services-based programming model. A processing workflow example is shown in Fig. 4. Its "data spaces" service takes a further step by providing explicit support for the kinds of data indexing and reorganization needed for code coupling [3]. The actions taken are in accordance with the resources provided to PreDatA, typically comprised of sets of nodes in a "staging area" reserved for PreDatA use by the application.

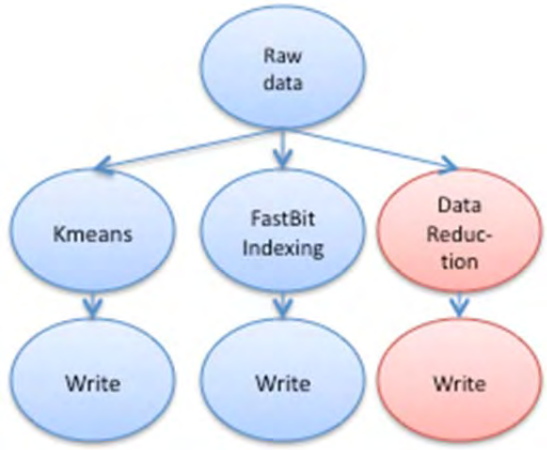


Figure 4. Workflow of three analysis pipelines. Red nodes have the highest priority that must be executed even if the application is blocked. Blue nodes can be skipped if there is no time for them.

The ADIOS framework, shown in Fig. 5, is an open source componentization of the I/O layer. It provides an easy-to-use programming interface that is uniform across multiple data formats. It enables the capture of metadata and provenance in the background without polluting the application software. In many cases, the performance of ADIOS is over 10x faster than other parallel output techniques. The second publically released version of the Adaptable I/O System [11] contains many new enhancements that allows users to use the PreDataA framework for I/O staging through the DataTap

[2] method, and through the NSSI [17] method. ADIOS 1.2 also contains a new method, MPI-AMR, which is designed to maximize the I/O performance on the Lustre file system. ADIOS has many different methods that allow users to gain near "peak" I/O performance for some of the largest applications running at the leadership class facilities of the Department of Energy [10] [16].

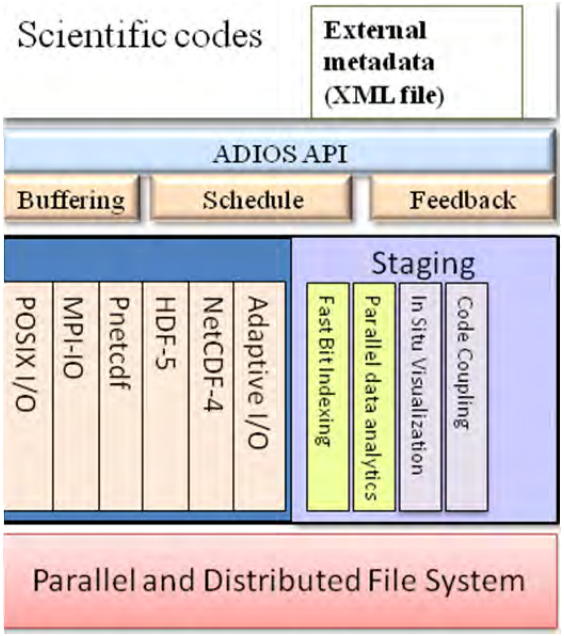


Figure 5. The ADIOS componentized I/O framework. The feedback mechanism and the services in the Staging area are proposed extensions.

4.3. Allow statistics to be added into files

ADIOS contains a new log-file format [13],that is self-describing, and contains a redundancy of metadata operations to allow the format to be both efficient in reading and writing, and resilient. The Binary-Packed (BP) format not only supports flexible conversion to standard file formats, such as NetCDF and HDF5, but it also facilitates a summary inspection of the data, termed data characteristics. ADIOS collects local, simple statistical and/or analytical data during the output operation for use in identifying desired data sets. The characteristics included in ADIOS 1.2 contain the minimum, maximum, average, standard deviation of an array at each time step. These characteristics have been tested from a variety of applications, and have been shown to take <4% of the total I/O cost of the simulation. Histograms are also included as an "optional" characteristic, where users can indicate the break points for the histogram of an output variable. When a user inquires about variability in an ADIOS-BP dataset, they can pull out these statistics without reading through the entire dataset. The bpls [17] utility can allow users to quickly look at large scale datasets, and determine these characteristics for all variables in seconds for extreme scale files.

4.4. In-memory code coupling

One of the methods inside the ADIOS framework is DataSpaces [6] which is an advanced coordination and interaction framework to provide the abstractions and mechanisms to support flexible and dynamic inter-application collaboration at runtime. DataSpaces uses Remote Direct Memory Access (RDMA) which is optimized for fast, asynchronous data transfers with low latency and small overheads. DataSpaces enables direct memory-to-memory communication between the nodes of distinct applications through RDMA. DataSpaces also enables the overlap of computations and communication allowing better utilization of the computing resources. By employing DataSpaces as a method in ADIOS, users can couple codes together with using I/O operations (open, write, read, close) as if writing into a file and reading from it.

The API used in ADIOS with the DataSpaces method, allows application scientists to change from file based coupling to in-memory coupling, with only changing the method used in ADIOS at runtime. The implementation consists of a client component that is integrated with the two application codes and allows for dynamic data exchange at run time, and a space component that runs in a set of dedicated "staging nodes", see Fig. 6. One of the advantages of using DataSpaces is that it allows users to use a Partitioned Global Address Space (PGAS) PGAS-like programming model.

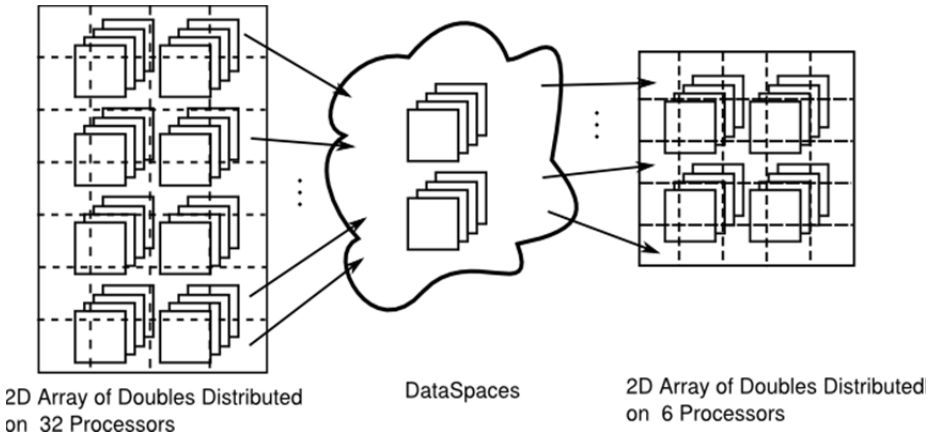


Figure 6. DataSpaces provides an abstract data space for exchanging data between applications

4.5. Web-based tools for monitoring, analysis, and data understanding

The Framework for Integrated End-to-end SDM Technologies (FIESTA) [4], has been used by several application groups to automate and monitor large scale simulations. One of the components of the FIESTA framework is the extendible, web-based, dashboard (eSimMon) [19] which is a front-end for simulation monitoring and analysis which helps scientists monitor, manage, visualize, and collaborate with other scientists. An example of the web-based screen is shown in Fig. 7. eSimMon uses Adobe Flash on the client side, and a combination of PHP, Python, and MySQL on the server side. The focus of using this technology, along with the provenance capturing system in the Kepler [1] workflow system, is to remove the data manipulation barrier from application scientists, such as file locations and formats, and allow them to focus on the science.

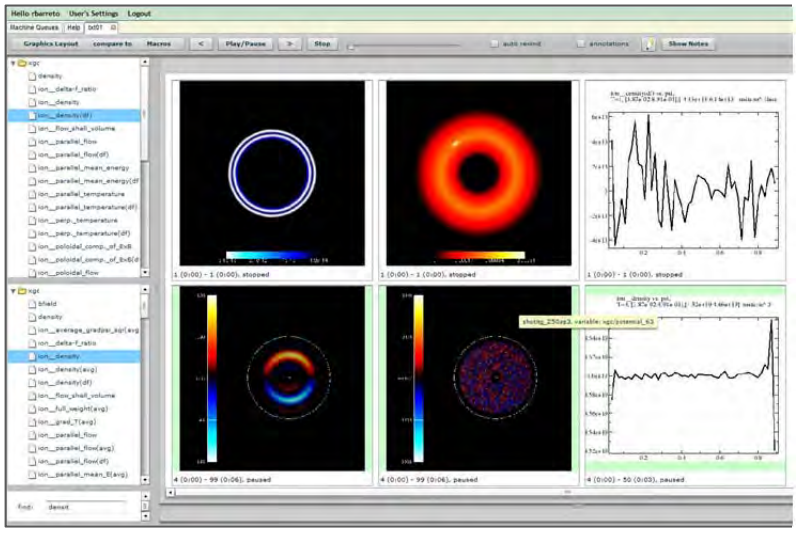


Figure 7. An example of a web-based dashboard for monitoring and evaluating simulations progress

On the back end, eSimMon uses PHP and the MySQL database to make the links between user requests on the interface and (BP, NetCDF, HDF5) data files. The Kepler workflow records provenance information in the database, which the dashboard can query to provide the linkage from the visualization on the dashboard and the actual data file. The recorded provenance includes the history of all of the data transformations, all of the operations that were executed, and the environment information, combined with the source code of the executed simulations and all actions of the users on the data. The provenance information is the key in enabling eSimMon to hide the details from the users and raise the focus from files to scientific variables.

The latest enhancement to eSimMon allows users to think of this environment as a Data Analysis Facility [TPK+10]. ESimMon now includes many analysis features, which can allow users to use statistical analysis packages, R and Matlab, run on a remote resource. Users can upload their Matlab

scripts, modify the input parameters, and share with their collaborators. These analysis services are run on analysis systems where the user uploaded them, and the underlying workflow technology of ESimMon can move data to the analysis service, or move the analysis to the data.

4.6. In-situ analysis in I/O or Staging nodes – a pipeline example

We have developed new technologies for managing the petascale resources used by high-end applications. Examples include scheduling methods that ensure that the I/O actions taken by HPC codes do not interfere with the codes' internal communications and in-memory support for code coupling via the increasingly common notion of "staging areas" on petascale machines. A further example is the ADIOS service-based approach for associating with HPC codes, which provides the functionality needed not only to output the enormous data volumes of HPC applications with high performance, but also to quickly analyze data and make it accessible to end users interested in scientific insights derived from their applications.

PreDataA adds to the current I/O stack data staging and in-transit processing capabilities. PreDataA resides on both the compute nodes, and the staging nodes, and dynamically decides on where different types of analysis can take place. PreDataA provides a framework, into which users can "plug-in" analysis or other data intensive processes. The advantage of using the PreDataA framework is that it can reduce the data being output to the file system, which becomes increasingly important as we move to the extreme scale.

We use two types of programming models inside our framework. The first is a stream processing programming model, which is shown in Fig. 8, which is similar to the map-reduce framework [5]. Each staging node is responsible for processing a stream of packed data chunks, with each chunk from separate compute processes. The processing of the streams is done in five stages, as shown in [20]. Together, this is similar to, but somewhat different than map-reduce, as described in [24]. By combining the PGAS-like programming model in DataSpaces, along with the PreDataA framework, we are exploring new opportunities in creating flexible programming models for users to "plug-in" their analysis, data reduction, visualization, and data indexing routines.

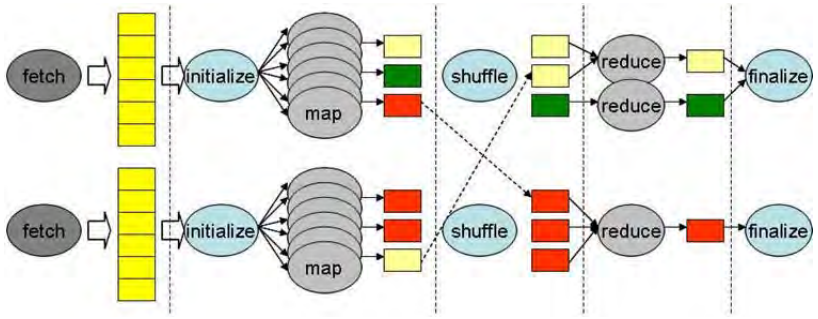


Figure 8. PreDatA streamlined processing pipeline

4.7. In-situ data summarization and reduction

In addition to performing analysis along the I/O pipeline, analysis and data reduction may also be performed on the nodes on which computation is occurring. This has the distinct advantage of allowing analysis routines to operate without the overhead of remote access to data, but requires that analysis routines scale to the number of nodes that the simulation is using and that these routines have methods for accessing the dataset in the simulation's native format(s).

Yu et al. provide a good example of this type of approach, integrating both volume and particle rendering into a combustion code running at as many as 15 thousand cores [23]. The approach is not limited strictly to performing analysis at runtime. In separate work, by selectively compressing the dataset itself Wang et al. [22] were able to reduce the amount of data stored for analysis by 20-30x while maintaining fidelity at regions of interest and the flexibility to choose and adjust parameters such as viewpoints and transfer functions after the fact. This is shown in Fig. 9, where the feature of interest in this case is the mixture fraction with an iso value of 0.2 (white surface), and the colored regions are a volume rendering of the HO2 variable.

4.8. Other methods to address I/O challenges

While much of the discussion here has covered software that sits between applications and the file system, parallel file systems themselves are in need of change. The hardware architecture of current HPC storage systems is not as cost effective as those used in data intensive computing environments [8], but HPC parallel file systems were not designed to operate reliably on faulty commodity components. Improvements in storage software will enable extreme scale storage solutions at a new price point. Likewise, the POSIX file system interface is increasingly hindering performance. New data models that deliver high productivity as well as high performance to computational scientists need to be supported throughout the I/O software stack, rather than being applied on top of this antiquated interface. It is all these approaches in concert, from in situ analysis to new and more effective data models that will enable us to meet the data volume and I/O challenges to come.

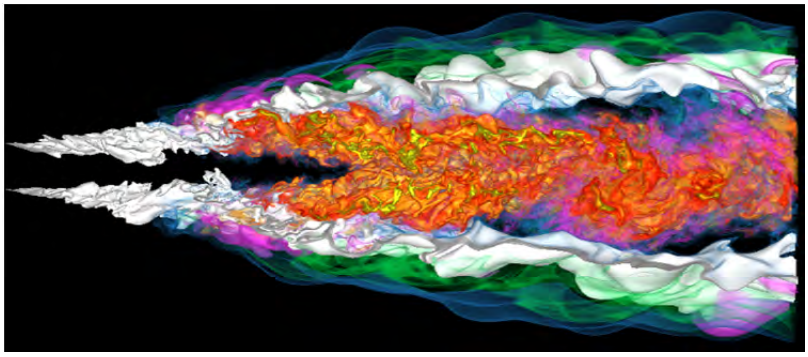


Figure 9. Stored values can be reduced significantly by showing details only for regions of interest. [Data courtesy J. Chen (SNL), image courtesy H. Yu (SNL)].

4.9. Indexing

In many scientific applications, to gain insight from massive amounts of data, the scientists have to locate a relatively small number of data records that hold the key information. For example, in a study of turbulent combustions, these special data records might be called ignition kernels. In a study of laser-wakefield particle accelerators, these special data records might be called particle bunches. Sifting through mountains of data to locate these "interesting" data records is a significant challenge. To meet this challenge, we have been developing and expanding an indexing software package called FastBit. An example showing parallel coordinate visualization display using FastBit for real-time selection of particle of interest (e.g. energetic particles) is shown in Fig. 10. This work, performed in collaboration with the Vacet Visualization center, replaced the existing IDL based analysis program

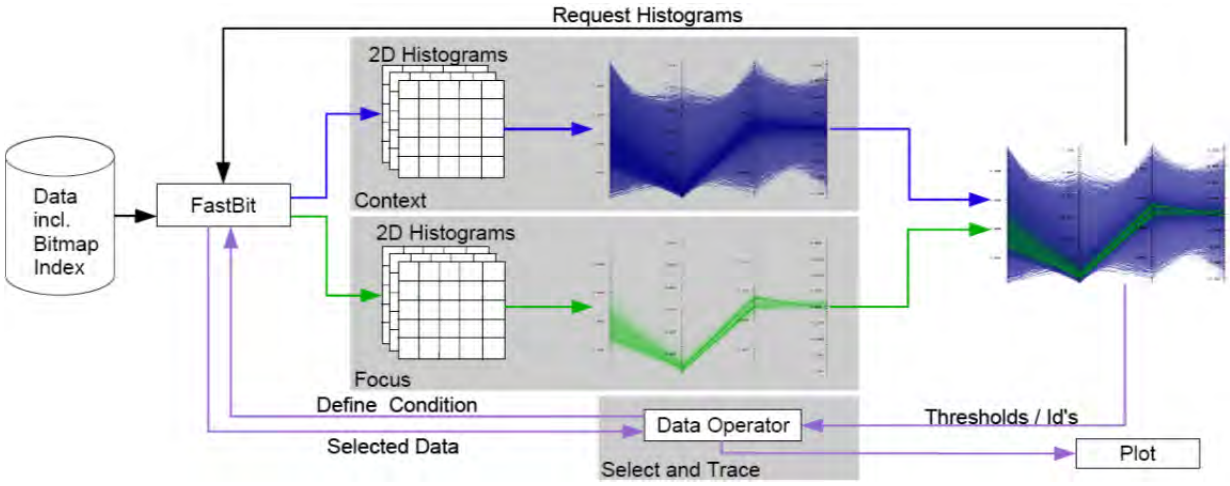


Figure 10. An example of query-driven visualization using FastBit indexes

with a FastBit based program, resulting in a three orders of magnitude speedup (from 300 seconds to 0.3 seconds), as reported in [18] for laser-wakefield particle accelerators, along with several other application areas.

FastBit is based on an indexing technique called bitmap index. This type of index is well-suited for searching scientific data where the data records usually remain unchanged after they are created. Taking advantage of this "read-only" (or "read-mostly") nature, FastBit indexes are designed to answer queries fast by sacrificing some efficiency in updating the indexes. This allows us to package the indexing data structures tightly, reduce the I/O requirement when answering a query and improve query response time. Additionally, FastBit is designed to work with user data in their existing formats, instead of demanding the user data to be transform into a particular format or loaded into a database management system. This flexibility makes it possible for users to accelerate their search operations with a minimal amount of change to their existing data analysis framework.

Efficient indexing of data is useful for both in-situ analysis and post-processing exploratory analysis. Having indexes available while the data is generated will allow in-situ monitoring, subset selection, and in-situ visualization in the extreme scale. For post-processing, generating the index while the data is generated saved the cost of generating the index after the data is saved on disk.

5. Summary

To address the ***data volume challenge,*** the main goal is to minimize the data to be stored. This can be done with several techniques: (1) perform in situ and co- analysis when possible, using extra cores, GPUs, I/O nodes, and staging nodes; (2) summarize data in-situ using parallel statistics methods, and enhance parallel algorithms to be performed by piece-wise statistical computation, taking advantage of multi-cores and GPUs; (3) avoid sending data to disk when it is only needed temporarily, such as data from uncertainty quantification (UQ) runs, when only summaries of each run are needed; (4) perform monitoring of simulation progress in situ, using staging nodes to support workflows (or pipelines); (5) keep details of data to be moved to disk only in regions of exploration, and summarize in other regions (as was shown in Fig.9).

To address the ***energy challenge***, the goal is to minimize data movement. Several techniques can be used: (1) store intermittent data, such as checkpoint data, on large NVRAM (non-volatile random access memory) rather than move to disk. Similarly, monitoring data can be stored on NVRAM, with appropriate tools for displaying results on web-based dashboards; (2) create indexes in-situ using extra core(s) to generate partial indexes, then combine results; (3) collect multiple "writes" and move them asynchronously when possible; (4) perform code-coupling in memory; (5) minimize data access after it is stored on disk – take advantage of indexing; (6) keep analysis operations as close to the stored data as possible; (7) continue to keep data that is needed long term on tape archives since they require no energy when not in use; (8) during post-processing exploratory analysis, bring from tape as needed using index information, power down disks when possible based on access patterns, or use multi-speed disks (a disk in "stand-by" mode requires only one-tenth of the energy of a full spin mode), and use NVRAM to front disks to hold frequently accessed files.

We believe that a combination of such techniques will be necessary to make future extreme scale machines effective and avoiding clogging them with unnecessary data.

References

[1] Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludäscher, B., and Mock, S., 2004, "Kepler: An Extensible System for Design and Execution of Scientific Workflows.

[2] Abbasi, H., Lofstead, J., Zheng, F., Klasky, S., Schwan, K., and Wolf, M., 2007, "Extending I/O through High Performance Data Services," IEEE International, Austin, TX.

[3] Abbasi, H., Wolf, M., Eisenhauer, G., Klasky, S., Schwan, K., and Zheng, F., 2009, "Datastager: Scalable Data Staging Services for Petascale Applications," ACM, Garching, Germany.

[4] Cummings, J., Lofstead, J., Schwan, K., Sim, A., Shoshani, A., Docan, C., Parashar, M., Klasky, S., Podhorszki, N., and Barreto, R., 2010, "Effis: An End-to-End Framework for Fusion

Integrated Simulation," Parallel, Distributed, and Network-Based Processing, Euromicro Conference on, 0(pp. 428-434.

[5] Dean, J., and Ghemawat, S., 2008, "Mapreduce: Simplified Data Processing on Large Clusters," Commun. ACM, 51(1), pp. 107-113.

[6] Docan, C., Parashar, M., and Klasky, S., 2010, "Dataspaces: An Interaction and Coordination Framework for Coupled Simulation Workflows," ACM, Chicago, Illinios.

[7] A. Geist, "Paving the Roadmap to EXASCALE," SciDAC Review, NUMBER 16 Special Issue 2010.

[8] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google File System. In Proceedings of the 19th ACM Symposium on Operating Systems Principles. pp. 29-43, 2003.

[9] Wei-keng Liao, and Alok Choudhary. Dynamically Adapting File Domain Partitioning Methods for Collective I/O Based on Underlying Parallel File System Locking Protocols. In the Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, Austin, Texas, November 2008.

[10] Lofstead, J., Fang, Z., Liu, Q., Klasky, S., Oldfield, R., Kordenbrock, T., Schwan, K., and Wolf, M., 2010, "Managing Variability in the Io Performance of Petascale Storage Systems," IEEE Press, New Orleans, LA.

[11] Lofstead, J., Klasky, S., Booth, M., Abbasi, H., Zheng, F., Wolf, M., and Schwan, K., 2009, "Petascale Io Using the Adaptable Io System," Cray User's Group.

[12] Jay Lofstead, Scott Klasky, Karsten Schwan, Norbert Pohorszki ,Chen Jin, 2008, "Flexible Io and Integration for Scientific Codes through the Adaptable Io System (Adios)," Proc. CLADE 2008 at HPDC, ACM, ed. Boston, Massachusetts.

[13] Lofstead, J., Zheng, F., Klasky, S., and Schwan, K., 2009, "Adaptable, Metadata Rich Io Methods for Portable High Performance Io," 2009 Ieee International Symposium on Parallel & Distributed Processing, Vols 1-5, pp. 778-787.

[14] Arifa Nisar, Wei-keng Liao, and Alok Choudhary. Scaling Parallel I/O Performance through I/O Delegate and Caching System. In the Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, Austin, Texas, November 2008.

[15] Kazuki Ohta, Dries Kimpe, Jason Cope, Kamil Iskra, Robert Ross, and Yutaka Ishikawa. Optimization techniques at the I/O forwarding layer. In Proceedings of the IEEE International Conference on Cluster Computing, September 2010 (to appear).

[16] Polte, M., Lofstead, J., Bent, J., Gibson, G., Klasky, S., Liu, Q., Parashar, M., Podhorszki, N., Schwan, K., Wingate, M., and Wolf, M., 2009, "...And Eat It Too: High Read Performance in Write-Optimized Hpc I/O Middleware File Formats.

[17] Podhorszki, N., Liu, Q., Lofstead, J., Klasky, S., Wolf, M., Schwan, K., and Team), T. A., 2010, Adios 1.2, 2010, http://users.nccs.gov/~pnorbert/ADIOS-UsersManual-1.2.pdf

[18] Oliver Ruebel, Sean Ahern, E. Wes Bethel, Mark D. Biggin, Hank Childs, Estelle Cormier-Michel, Angela DePace, Michael B. Eisen, Charless C. Fowlkes, Cameron G.R. Geddes, Hans Hagen, Bernd Hamann, Min-Yu Huang, Soile V.E. Keraenen, David W. Knowles, Cris L. Luengo Hendriksl, Jitendra Malik, Jeremy Meredith, Peter Messmer, Prabhat, Daniela Ushizima, Gunther H. Weber and Kesheng Wu. Coupling visualization and data analysis for knowledge discovery from multi-dimensional scientific data Procedia Computer Science Volume 1, Issue 1, May 2010, Pages 1751-1758 ICCS 2010

[19] Santos, E., Tierny, J., Khan, A., Grimm, B., Lins, L., Freire, J., Pascucci, V., Silva, C. T., Klasky, S., Barreto, R., and Podhorszki, N., 2009, "Enabling Advanced Visualization Tools in a Web-Based Simulation Monitoring System," e-Science and Grid Computing, International Conference on, 0(pp. 358-365.

[20] R. Tchoua, S. Klasky, N. Podhorszki, B. Grimm, A. Khan, E. Santos, C. Silva, P. Mouallem, and Vouk, M., 2010, "Collaborative Monitoring and Analysis for Simulation Scientist," Proc. The 2010 International Symposium on Collaborative Technologies and Systems (CTS 2010), Chicago, IL.

[21] Venkatram Vishwanath, Mark Hereld, Kamil Iskra, Dries Kimpe, Vitali Morozov, Michael E. Papka, Robert Ross, and Kazutomo Yoshii. Accelerating I/O forwarding in IBM Blue Gene/P systems. In Proceedings of Supercomputing, November 2010 (to appear).
[22] Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma. Application-driven compression for visualizing large-scale time-varying volume data. IEEE Computer Graphics and Applications, 2009.
[23] Hongfeng Yu, Chaoli Wang, Ray Grout, Jacqueline Chen, and Kwan-Liu Ma. In Situ Visualization for Large-Scale Combustion Simulations. In IEEE Computer Graphics and Applications, Vol. 30, Number 3, May/June 2010.
[24] Zheng, F., Abbasi, H., Docan, C., Lofstead, J., Klasky, S., Liu, Q., Parashar, M., Podhorszki, N., Schwan, K., and Wolf, M., 2009, "Predata-Preparatory Data Analytics on Peta-Scale Machines," Proc. 24th IEEE International Parallel and Distributed Processing Symposium, Atlanta.

Three-dimensional full core power calculations for pressurized water reactors

T.M. Evans,[1] G.G. Davidson,[1] and R.N. Slaybaugh[2]

[1]Radiation Transport Group, P.O. Box 2008, Oak Ridge National Laboratory, Oak Ridge, TN 37831
[2]Engineering Physics Department, University of Wisconsin-Madison, 147 Engineering Research Bldg., 1500 Engineering Drive, Madison, WI 53706

Email: evanstm@ornl.gov

Abstract. We have implemented a new multilevel parallel decomposition in the Denovo discrete ordinates radiation transport code. In concert with Krylov subspace iterative solvers, the multilevel decomposition allows concurrency over energy in addition to space-angle. The original space-angle partitioning in Denovo placed an effective limit on the scalability of the transport solver that was highly dependent on the problem size. The added phase-space concurrency combined with the high-performance Krylov solvers has enabled weak scaling to $O(100K)$ cores on the Jaguar XT5 supercomputer. Furthermore, the multilevel decomposition provides sufficient concurrency to scale to exascale computing and beyond.

1. Introduction

Predictive nuclear energy simulations will involve the coupled modeling of many physical regimes, including Boltzmann transport, and will require tremendous computational resources. Experience in the Department of Energy (DOE) Advanced Simulation and Computing program, astrophysics, and nuclear energy have demonstrated that, in coupled-physics calculations, a 3-D Boltzmann transport solver will generally require the vast majority of computational resources, in terms of both memory and operations, because of the seven-dimensional phase-space (location, velocity [energy + angle], and time). For a nuclear reactor simulation, the scale of the problem is large: 5 orders of magnitude in space and 10 in neutron energy. A transport solver that incorporates a resolved discretization for all scales using current discrete models would require 10^{17-21} degrees of freedom (DOF) for a single timestep, which is beyond even exascale computational resources. This problem size precludes, for the time being, a single integrated ab initio computational approach. Instead, variations of current multilevel techniques will continue, with the new objective being to make the process more consistent and, subsequentially, more predictive.

Present reactor transport methods use an *inconsistent* three-level homogenization approach, often utilizing distinct simulation codes, in modeling radiation transport in the core of a nuclear reactor. Figure 1 shows the spatial and energy domains, respectively, of this multiscale challenge: α, use of a fine mesh in 1-D cylindrical geometry of an approximate small subset (pincell) of the reactor core with a first-principles representation of the energy spectrum; β, use of a coarser mesh with a 2-D transport solution in a larger subset (lattice) of the core with grouped representation of the energy spectrum provided by the previous step; and γ, use of a very coarse mesh in a 3-D diffusive transport of neutrons in the full homogenized core of the reactor with a very coarse representation of the energy spectrum

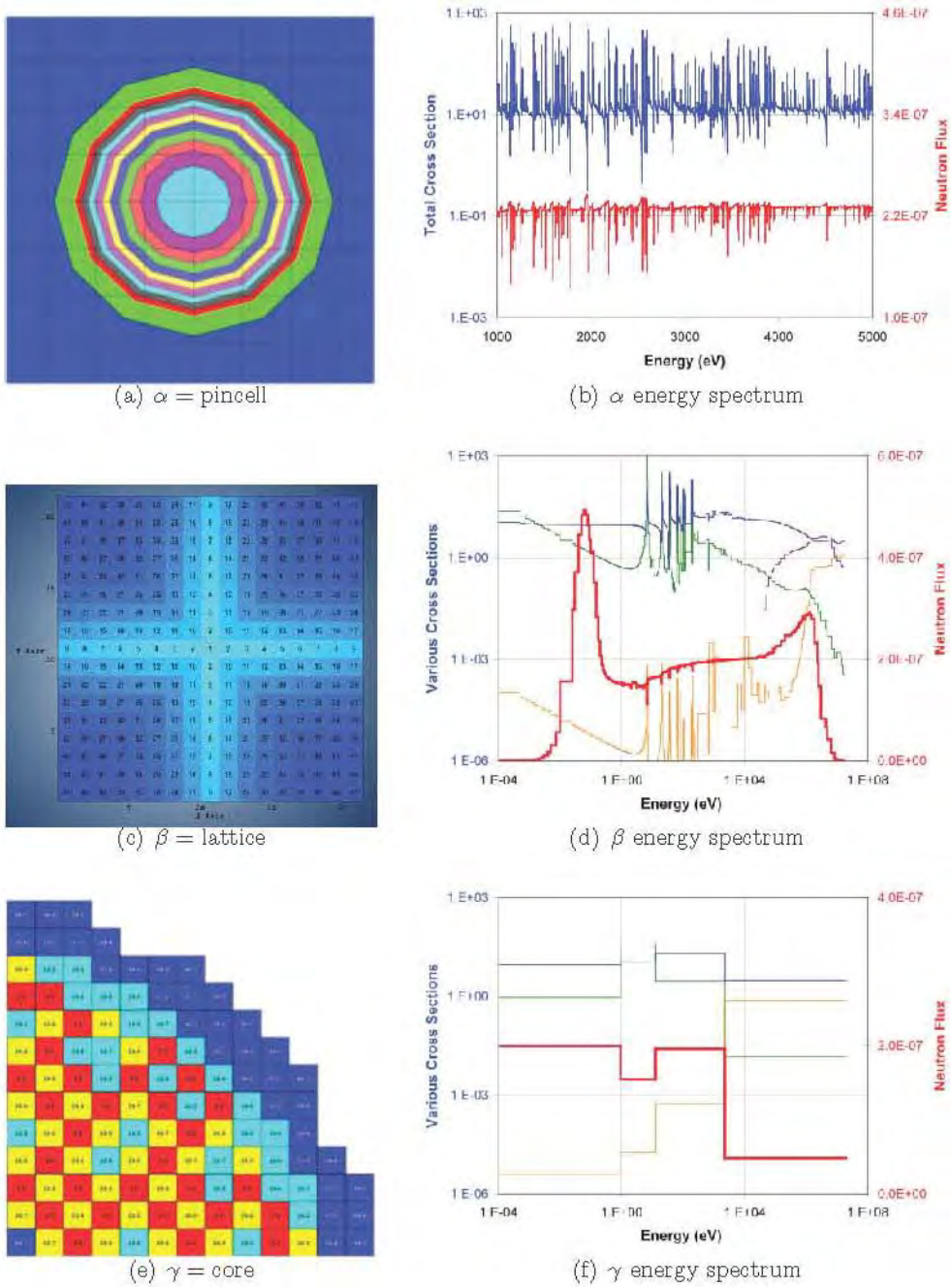


Figure 1. Three levels of reactor geometries and energy structures.

provided by the previous step. The first two steps (α, β) require 10^{7-8} DOF with 10^{2-3} independent calculations, each on single-processor machines. Recent work at Oak Ridge National Laboratory (ORNL) has demonstrated that, with moderate computational resources, this can be reduced to an *inconsistent* two-step approach, where step (A) utilizes the energy fidelity of α with the spatial domain of β and step (B) uses the energy fidelity of β and the spatial domain of γ [1,2]. In this approach, each step would require 10^{11-13} DOF per step with 10^2 independent high-order (A) calculations for every timestep.

The solution of the first-order form of the Boltzmann transport equation requires multiple wavefront solutions over each discrete angle. The wavefront solver imposes a fundamental limitation on the

scalability of the algorithm because of the upstream dependencies of downstream regions. Various approaches to this problem over the last decade have yielded mixed results. The Koch-Baker-Alcouffe (KBA) [3] algorithm for structured 3-D grids is very efficient in that the transport operator can be inverted in a single solve. However, the scalability is limited by communication latency such that this algorithm is limited by problem size and generally will not scale to $O(100K)$ cores. Parallel-Block-Jacobi methods have been applied to both structured and unstructured grids yielding excellent weak-scaling results [4]. But, these methods are less efficient in terms of memory and iteration count when compared with direct wavefront algorithms. Graph-based direct methods have been investigated on unstructured grids, but these algorithms have significant scaling limitations [5].

To reduce today's three-level approach to consistent single- or two-level schemes, the orders-of-magnitude increase in fidelity of each of the steps will require substantially more computational resources than present algorithms utilize. Therefore, novel approaches to parallelize the transport equation must be developed that can overcome the wavefront limitation and take full advantage of the complete computational resources.

A parallel decomposition over energy groups and an Arnoldi-based k-eigenvalue solver have been developed for the Denovo code. Denovo is a 3-D discrete ordinates (S_N) multigroup radiation transport code for radiation shielding and reactor physics applications under active development at ORNL [6.7]. The new energy decomposition is in addition to the KBA-based spatial domain decomposition already present in Denovo. This multilevel decomposition allows parallel scaling up to hundreds of thousands of cores. Additionally, the new Arnoldi solver can solve k-eigenvalue problems with many fewer mesh sweeps than the traditional power iteration method.

In this paper we will describe Denovo's multilevel parallel decomposition and the solver technologies that complement it. Solvers for the multigroup S_N equation are described in § 2. In § 3 we show the limitations of KBA space-angle parallelism on leadership class platforms. Finally, we describe a multilevel parallel decomposition that overcomes these shortcomings in § 4, and in § 5 we summarize the complete set of solver options available in Denovo. Results that show the efficiency of these methods are shown in § 6.

2. Multigroup Solvers

To see how a multilevel energy decomposition can be beneficial and how it will be implemented, we briefly review the fundamental solver strategies in Denovo; Ref. [6] can be consulted for full details. The multigroup S_N equations can be written in operator form as

$$\mathbf{L}\psi = \mathbf{MS}\phi + q_e\,, \quad \text{(fixed source)} \tag{1}$$

$$\mathbf{L}\psi = \mathbf{MS}\phi + \frac{1}{k}\mathbf{M}\chi f^T\phi\,. \quad \text{(eigenvalue)} \tag{2}$$

The state of these equations is defined in angular flux moments, ϕ, that are related to the discrete angular flux through

$$\phi = \mathbf{D}\psi\,, \tag{3}$$

where $\mathbf{D}$ is the discrete-to-moment operator that integrates discrete angles into angular flux moments using quadrature rules. $\mathbf{L}$ is the first-order linear differential transport operator, $\mathbf{M}$ is the moment-to-discrete operator that projects angular flux moments into discrete angle space, and $\mathbf{S}$ is the group-to-group scattering matrix. In the eigenvalue form of the equation, $\mathbf{f}^T$ is the row vector of fission cross sections, χ is the fission spectrum vector, and k is the largest eigenvalue.

Operating by $\mathbf{DL}^{-1}$, defining $\mathbf{T} = \mathbf{DL}^{-1}$, and rearranging terms, the fixed-source problem is

$$(\mathbf{I} - \mathbf{TMS})\phi = q\,, \tag{4}$$

where $q = \boldsymbol{T}q_e$. Similarly, the eigenvalue problem becomes

$$(\mathbf{I} - \mathbf{TMS})\phi = \frac{1}{k}\mathbf{TMF}\phi\,, \tag{5}$$

where $\mathbf{F} = \chi f^T$ is the full-rank fission matrix. The operator $\mathbf{L}^{-1}$ can be formed into a lower-triangular system if one sweeps the space-angle grid in the direction of neutron travel. The resulting transport *sweep* is the operation that is parallelized using the KBA algorithm. This matrix is never formed in practice; only the action of the operator on a vector, $y = \mathbf{L}^{-1}v$, is required.

The traditional method for solving Eq. (4) is Gauss-Seidel iteration,

$$(\mathbf{I} - TMS_{gg})\phi_g^{k+1} = q_g + TM(\textstyle\sum_{g'=0}^{g-1} S_{gg'}\phi_{g'}^{k+1} + \sum_{g'=g+1}^{G} S_{gg'}\phi_{g'}^{k}) \,, \tag{6}$$

which is the same as solving G one-group space-angle problems per Gauss-Seidel iteration. We refer to these one-group problems as *within-group* equations, and they have the following form:

$$(\mathbf{I} - TMS_{gg})\phi_g = \bar{q}_g \,, \tag{7}$$

where $\bar{q}_g$ is an effective group source that includes all in-scattering, fission, and/or external sources for the group. When using Gauss-Seidel iteration over energy, the within-group solves represent a set of inner iterations. Denovo provides several solvers (Krylov, source iteration) for the within-group equations, and Diffusion Synthetic Acceleration (DSA) is available as a preconditioner for the Krylov options. Obviously, when $\mathbf{S}$ is lower triangular (indicating downscattering only) the solution converges in one Gauss-Seidel iteration. However, when $\mathbf{S}$ is energy dense, the solution can converge very slowly. This structure occurs in the low-energy region of the spectrum where neutrons undergo Maxwellian upscattering. A typical $\mathbf{S}$ for a 27-group set of common reactor materials is illustrated in Figure 2.

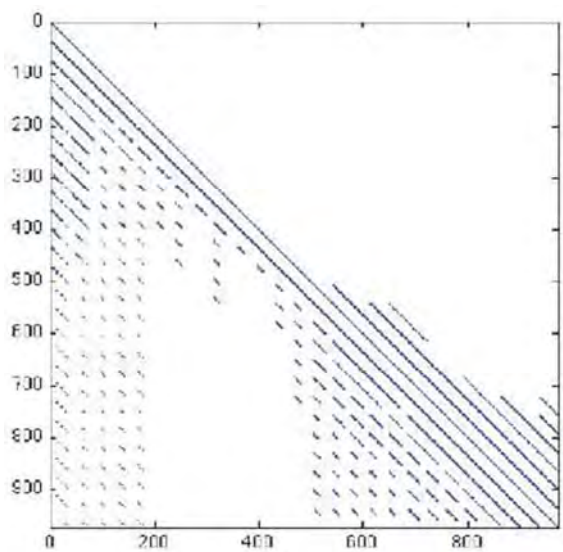


Figure 2. Sparsity plot of the scattering matrix for a problem containing iron, graphite, and heavy water. The data has 27 energy groups, and the matrix is lower-triangular through group 14. The full matrix is dimensioned over energy-space-angle with angle represented by Legendre moment expansion.

The standard way to solve Eq. (5) is power iteration. We start by defining an energy-independent eigenvalue,

$$\Gamma = f^T\phi \,. \tag{8}$$

Then, Eq. (5) can be written

$$A\Gamma = k\Gamma\,, \tag{9}$$

with

$$A = f^T(I - TMS)^{-1}TM\chi\,. \tag{10}$$

Solving by power iteration proceeds as follows

$$\Gamma^{k+1} = \frac{1}{k}A\Gamma^k\,. \tag{11}$$

The operation $A\Gamma^k$ necessarily involves solving multigroup problems with the same form as Eq. (4),

$$(I - TMS)y^k = TM\chi\Gamma^k\,. \tag{12}$$

Both Gauss-Seidel iteration for fixed-source problems and power iteration (with Gauss-Seidel inners) are provided in the current version of Denovo. While the implementation in Denovo is enhanced by applying Transport Two-Grid (TTG) acceleration to the Gauss-Seidel iterations [6] and using GMRES(m) on the inner one-group solves, a fundamental limitation of these solvers is that they are parallelizable only over space-angle variables. The recursive nature of Gauss-Seidel prevents effective parallelization over energy. One could use parallel block-Jacobi iteration, but this option results in solvers that are too inefficient to be of practical use on most problems.

Instead, the Krylov solver framework in Denovo that is currently used in the inner one-group space-angle solves has been expanded to include energy. Including energy in the Krylov vectors enables the following benefits:

- The energy variable is decoupled allowing groups to be solved independently.
- Krylov subspace iteration is more efficient and robust than Gauss-Seidel iteration.
- Preconditioning a Krylov iteration is generally more robust and stable than Gauss-Seidel acceleration.

Furthermore, including energy in the Krylov vector does not invalidate any of the existing sweep mechanics that are already implemented in Denovo.

For multigroup fixed-source problems in the form of Eq. (4), application of a Krylov method requires the following two steps:

(i) A full energy-space-angle sweep of the right-hand side source,

$$q = T\hat{q}\,, \tag{13}$$

where $\hat{q}$ is an effective source that could be an external source (q_e) in the case of true fixed-source problems, or it could be a fission source iterate when nested inside power iteration.

(ii) A full energy-space-angle sweep each Krylov iteration to calculate the action of the operator on the latest iterate,

$$y^k = (I - TMS)v^k\,, \tag{14}$$

where v^k is the Krylov vector in iteration k. We note that this vector is dimensioned $v^k \equiv \{v^k_{g,c,n,l,m}\}$ where g is the energy group, c is the cell index, n is the spatial unknown index in the cell, and (l, m) are the spherical harmonic moment indices.

For eigenvalue problems we have implemented an Arnoldi Krylov subspace solver using the Trilinos Anasazi package [8] that can (1) take full advantage of the energy parallelism and (2) be more efficient than power iteration. Arnoldi iteration requires the eigenproblem to be written in standard form,

$$Ax = \lambda x\,. \tag{15}$$

Arnoldi iteration can implemented with either an energy-dependent or energy-independent eigenvector as follows:

$$2A\phi = k\phi\,, \quad A = (I - TMS)^{-1}TMF\,, \quad \text{(energydependent)} \tag{16}$$

$$A\Gamma = k\Gamma\,, \quad A = f^T(I - TMS)^{-1}TM\chi\,. \quad \text{(energyindependent)} \tag{17}$$

In either case, the implementation of Arnoldi iteration requires a matrix-vector multiply at each Krylov iteration of the form

$$y^k = Av^k\,. \tag{18}$$

For the energy-dependent case we have

$$z^k = TMFv^k\,, \quad \text{(matrix} - \text{vectormultiplyandsweep)} \tag{19}$$

$$(I - TMS)y^k = z^k\,. \quad \text{(fixed} - \text{sourcesolve)} \tag{20}$$

Similarly, for the energy-independent eigenvector the steps are

$$z^k = TM\chi v^k\,, \quad \text{(matrix} - \text{vectormultiplyandsweep)} \tag{21}$$

$$(I - TMS)y^k = z^k\,, \quad \text{(fixed} - \text{sourcesolve)} \tag{22}$$

$$y^k \leftarrow f^T y^k\,. \quad \text{(dot} - \text{product)} \tag{23}$$

Both methods require a fixed-source solve each iteration. We consider both the energy-dependent and independent approaches because we are uncertain a priori which method will be optimal for a given problem. The energy-dependent approach allows parallelization of the eigenvalue solve across energy at the expense of a much larger eigenvector. The energy-independent approach allows only energy-domain parallelization over the fixed-source solve, and the eigenvalue solve is parallel only over space-angle. However, this decomposition may be more efficient because the eigenvector is smaller, especially when work is dominated by the inner multigroup fixed-source solve.

3. Sweep-Based Parallelism

Denovo [6] uses the KBA wavefront algorithm to parallelize the space-angle transport sweeps shown in Eqs. (4) and (5). Unfortunately, KBA alone provides insufficient parallelism on very large systems. The following analysis will demonstrate its shortcomings. The theoretical efficiency of KBA, ignoring machine latency, is

$$\varepsilon_{\max} = \frac{2MK}{2MK + K_b(I/I_b + J/J_b - 2)} = \frac{2MB_K}{2MB_K + P_I + P_J - 2}\,, \tag{24}$$

where (I, J, K) are the number of cells in (x, y, z), respectively. The number of cells per domain in (x, y) is given by (I_b, J_b), and K_b is the number of cells per on-processor block in the z dimension. Then, P_I and P_J are the number of processors in the x and y directions, respectively; B_K is the number of blocks in the z direction on each domain; and M is the number of angles per octant.

We have done strong scaling studies on Jaguar* with a $400 \times 400 \times 400$ cell mesh. The results are shown in Figure 3. Figure 3a shows that while a very high ($> 90\%$) maximum theoretical parallel efficiency is estimated, this value is impossible to achieve in practice. To get high theoretical efficiencies, the number of cells per block ($I_b \times J_b \times K_b$) becomes very small. This makes sense in the abstract because work is passed to subsequent blocks more rapidly. However, in reality the latency per message quickly overwhelms the theoretical prediction. When the number of cells per block becomes very small, data spend more time waiting in MPI message queues than working to solve the block. This effect is illustrated in Figure 3b. The deviation between the theoretical prediction and the measured efficiency becomes small as the block size grows. In summary, high theoretical efficiencies requiring small numbers of cells per block cannot be achieved, but theoretical efficiencies that result from larger block sizes can be realized.

* Jaguar XT5 supercomputer at the Oak Ridge Leadership Computing Facility (OLCF).

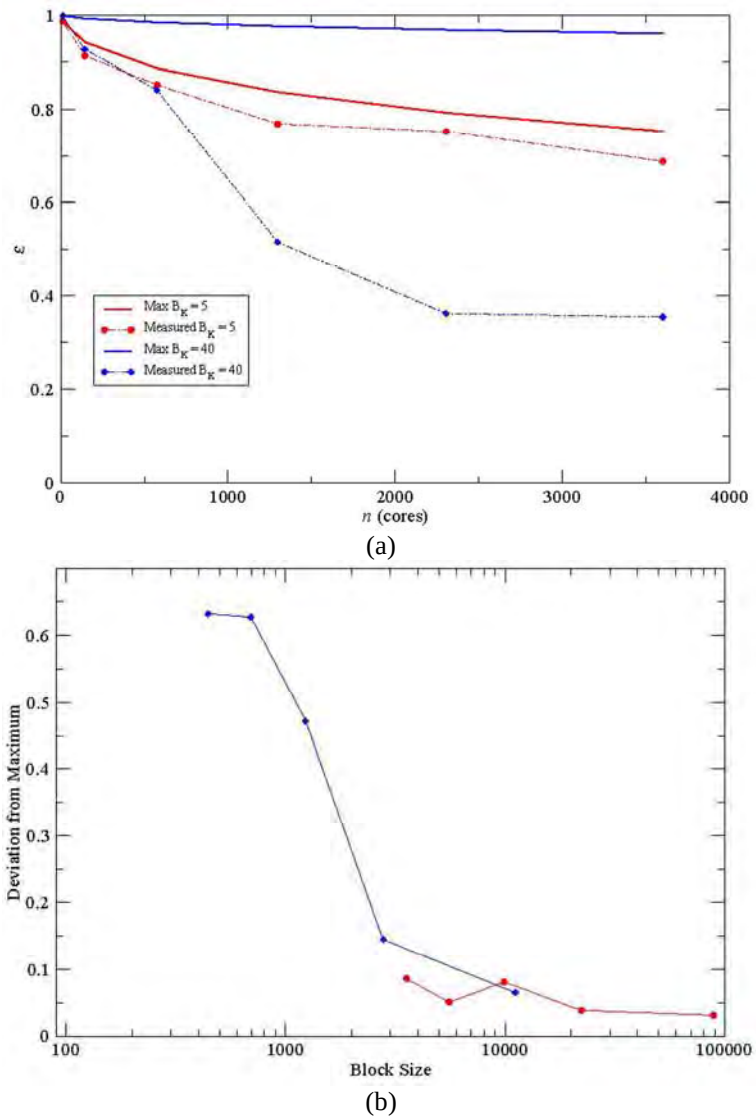


Figure 3. Denovo strong scaling results on Jaguar; (a) strong scaling with $B_K = 40$ blocks (blue) and $B_K = 5$ blocks (red) and (b) the deviation from the theoretical maximum as a function of number of cells per block.

The repercussions from this analysis are obvious; namely, the full extent of computational resources available on Jaguar cannot be effectively utilized. The best efficiencies are obtained when the block size can be set greater than ~1500. Thus, for any given problem, the maximum number of cores is predetermined by the minimum block size. Even a 500M cell problem will be limited to 15,000–20,000 cores under these restrictions. To utilize the full resources of Jaguar we must find additional variables to parallelize. Using the advanced solvers in Denovo, a multilevel decomposition over energy will provide the necessary parallelism.

4. Multilevel Parallel Decompositions

Having described the multigroup solvers in § 2 and discussed the limitations of parallelizing only the space-angle sweep, we now explain the parallel implementation of Denovo's energy-space-angle decomposition. The multilevel energy-space decomposition is illustrated in Figure 4. In this

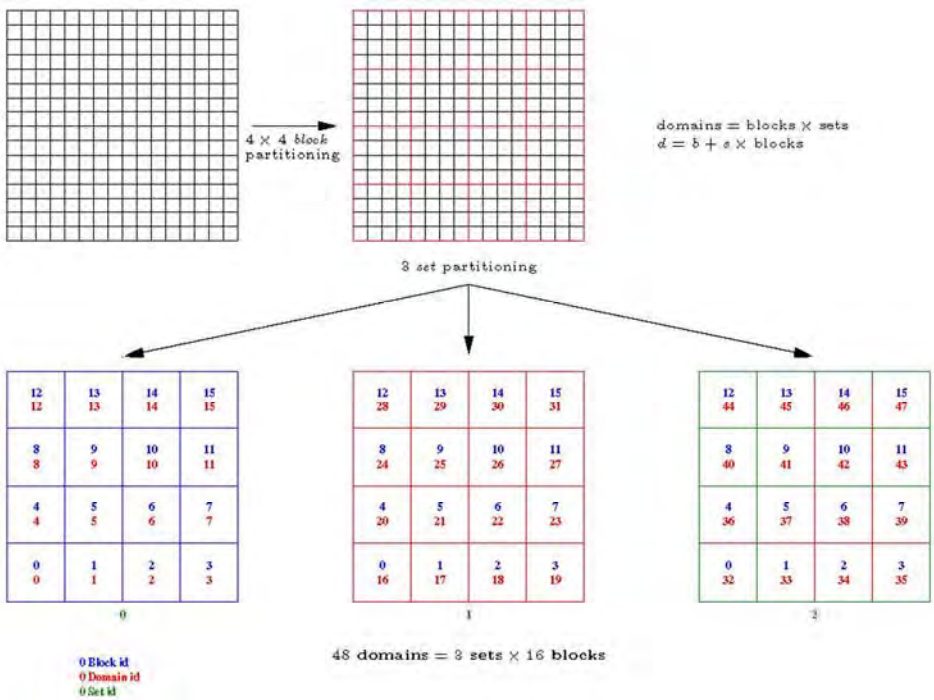


Figure 4. Multilevel energy-space decomposition in Denovo. This example has 3 *sets*, each containing 16 *blocks*, resulting in 48 total *domains*. The block (blue), set (green), and domain (red) IDs are indicated by b, s, and d, respectively.

decomposition, space is partitioned into *blocks*. Energy is partitioned into *sets*. Each set contains the full mesh (all of the blocks) such that KBA sweeps never cross set boundaries. Every (block, set) combination is termed a *domain*. The total number of domains is currently the same as the number of MPI processes in a parallel job. The old Denovo space-angle decomposition can be thought of as a single-set energy decomposition over $P_I \times P_J$ blocks. A benefit of this energy-space partitioning is that all of the solvers described in § 2 can be implemented in the new decomposition using Denovo's existing space-angle sweep machinery.

To solve the multigroup equations, Eq. 14 implies a matrix-vector multiply of the form

$$s_g = S_{g0} v_0 + S_{g1} v_1 + \cdots + S_{gG} v_G \,. \tag{25}$$

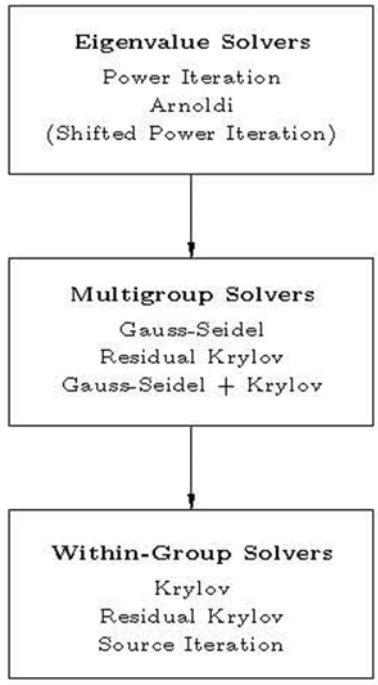
Instead of communicating all of the groups to each set so that the matrix-vector multiply can be completed locally, we replicate the source vector s. Then, each set performs its part of the matrix-vector multiply followed by a global reduction. The global reduction is performed using a communicator that relates all blocks with the same index across sets. After the global reduction, each set has the complete source vector s, even though it will utilize only the components of s that are local to the set.

After calculating the sweep source s, the sweeps for each local group on a set can be performed without any interset communication. The sweeps require only communication between blocks within a set. The only exception to this rule is when the energy-independent version of Arnoldi is applied. In this case, the eigenvector must be summed across all sets in a manner analogous to that described above for calculating the sweep source.

5. Denovo Solver Taxonomy

Having reviewed the new solvers and parallel decompositions in Denovo, we will now summarize the complete set of solver options that are available in the code. We have separated the taxonomy into

Within-Group Solvers, *Multigroup Solvers*, and *Eigenvalue Solvers*. These are arranged in levels according to the following:



Fixed-source problems are solved using multigroup solvers.

5.1. Within-Group Solvers

The within-group solvers are used to solve Eq. (7). All of the within-group solvers are parallelized over space because, by definition, they require no coupling between energy groups. Thus, they operate only within a set, not across sets.

Solver	Preconditioning	Parallelization
Direct Krylov	DSA	Interblock KBA
Residual Krylov	DSA	Interblock KBA
Source Iteration		Interblock KBA

5.2. Multigroup Solvers

The multigroup solvers are used to solve Eq. (4). They can be used independently to solve fixed-source problems, or they can be used in the inner iterations of eigenvalue problems. The multigroup solvers are parallelized over space-angle (blocks) and energy (sets), although not all solvers support energy parallelization. For example, the Gauss-Seidel solver does not support parallelization over energy.

Solver	Preconditioning	Parallelization
Gauss-Seidel	TTG	Single-set energy partitioning
Gauss-Seidel/Krylov		Gauss-Seidel over downscatter groups replicated on each set, Krylov iteration over upscatter groups using multiset energy partitioning
Krylov	Multigrid Energy, LU*	Multiset energy partitioning

*In development.

5.3. Eigenvalue Solvers

The eigenvalue solvers solve Eq. (5). The parallelization is largely determined by the choice of multigroup solver. Some eigenvalue solvers can solve both energy-dependent and energy-independent eigenvectors, and this choice dictates the parallelization strategy.

Solver	Eigenvector	Multigroup Solvers
Power Iteration	Energy independent	Gauss-Seidel, Krylov, and Gauss-Seidel/Krylov
Arnoldi	Energy independent/ energy dependent	Krylov and Gauss-Seidel/Krylov
Rayleigh Quotient Iteration	Energy dependent	Krylov

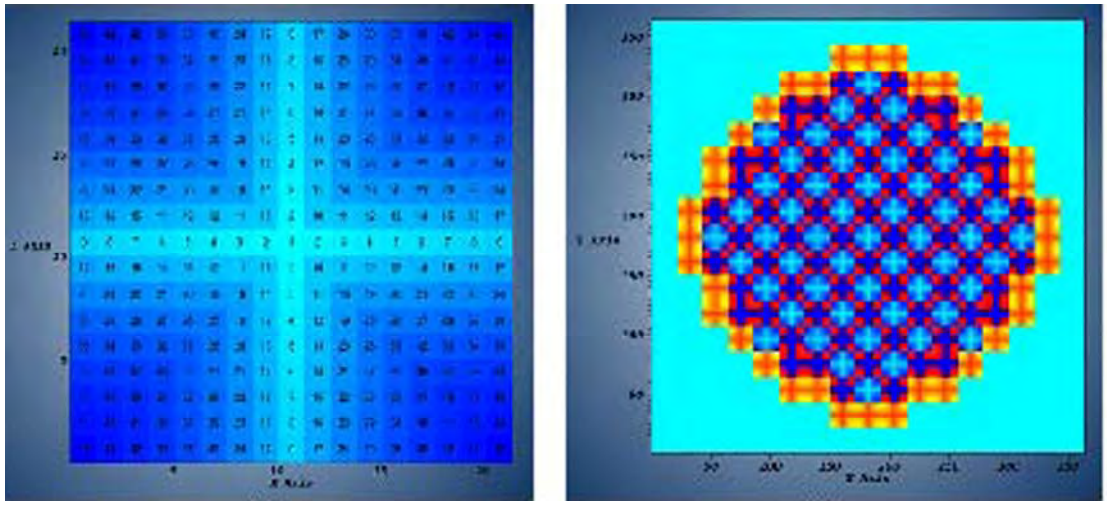
6. Results

Each of the solvers and parallel algorithms has been independently verified through the Denovo test suite. To test the performance on leadership-class computing hardware, we have choosen a generic whole-core pressurized water reactor (PWR) model as a test problem. This model was originally developed as a whole-core, pin-homogenized, two-group benchmark problem by Électricité de France (EDF) working with the University of Florida [9]. Although the ultimate objective is to perform whole-core, pin-resolved, 3-D transport simulations, a pin-homogenized problem serves as a useful and realistic benchmark problem. Denovo is used to solve Eq. (5) in order to calculate the k-eigenvalue and the scalar flux throughout the core. Using the scalar flux, the pin power distribution, fission source, and group-wise power distributions can be analyzed. The ability to solve pin-homogenized, whole-core problems with transport, as opposed to diffusion or other low-order approximations, is the first step towards fully predictive reactor core modeling and simulation. To achieve first-principles predictive capability, each pin would be fully resolved in the whole-core 3-D model. This objective cannot be approached until we have demonstrated the ability to solve pin-homogenized 3-D reactor problems with full transport.

The model is a generalization of a Westinghouse PWR-900 core that has a core height of 4.2 m, an assembly height of 3.6 m, and a lattice pitch of 1.26 cm. The core features 289 assemblies, of which 157 are fuel and 132 are in the reflector. Each assembly contains a 17×17 array of homogenized fuel pins that are arranged with $1/4$ lattice symmetry as shown in Figure 5a. Three different fuel enrichments ranging from 1.5% to 3.25% are used in the assemblies. We implemented this model in the TRITON sequence of SCALE [10] to generate 44-group pin-homogenized cross sections to supplement the two-group set defined in the benchmark. Each set of pin-homogenized cross sections contains 135 unique materials (45 pins at 3 levels of enrichment). The 2-D radial view of the core is shown in Figure 5b.

For Denovo, the model was discretized into $2 \times 2 \times 700$ spatial cells per homogenized fuel pin, resulting in a total mesh size of 233,858,800 cells. An angular quadrature containing 168 angles, P_0 scattering (one angular moment), and step-characteristics spatial differencing was used for all calculations. These parameters yielded 39,288,278,400 DOF per energy group. All calculations were performed on the Jaguar XT5 computer at the OLCF.

Table 1 compares the different solvers for the two-group version of the PWR-900 benchmark. The 17,424 spatial domains used for the standard Gauss-Seidel (GS) solver represent the maximum number of cores that could be effectively used for this problem. To use more computing resource the multilevel decomposition is required. These results show that the multigroup Krylov solvers are very effective, and they allow an efficient multilevel decomposition. However, because this is a coarse energy benchmark, we cannot make dramatic conclusions about the performance of the solvers on real problems of interest.



(a) (b)

Figure 5. (a) PWR-900 17×17 pin fuel assembly. The pins have been homogenized into 45 unique materials in each assembly. All assembly enrichments have the same $1/4$ symmetry pattern. (b) 2-D radial cut of the reactor core. The low-enrichment assemblies are light blue, the medium enrichment assemblies are red/blue, and the high-enrichment assemblies are yellow/orange.

Table 1. Solver comparisons for 2 group version of the PWR-900 benchmark problem. In each case the eigenvalue was converged to 1-3. Each problem had 78,576,556,800 DOF.

Solver	Blocks	Sets	Domains	Solver Time (min)
PI/GS + TTG	17,424	1	17,424	11.00
PI/GMRES	10,200	2	20,400	3.03
Arnoldi/GMRES	10,200	2	20,400	2.05

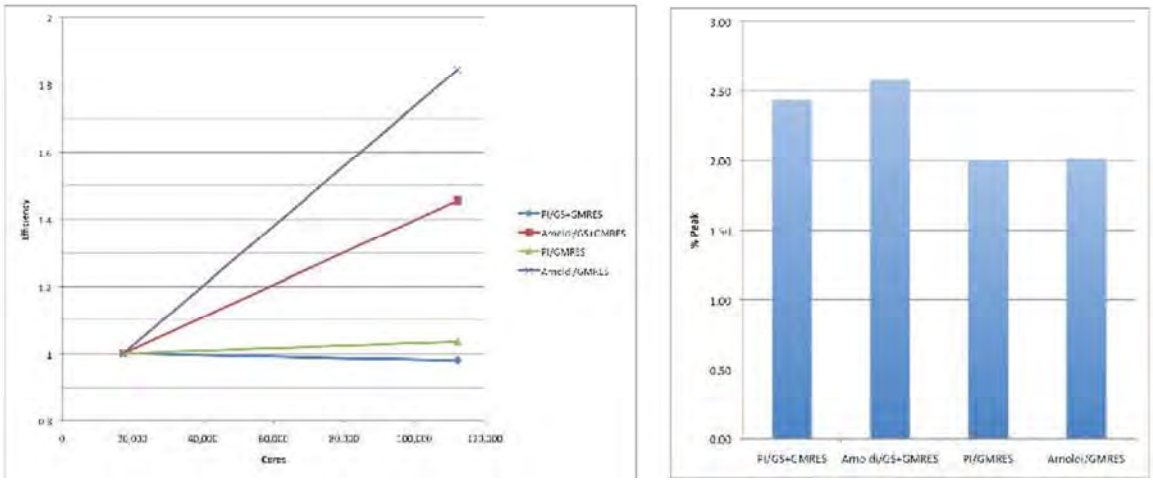
To show the validity of this approach for fully resolved reactor problems, we must investigate the performance of the parallel algorithm and solvers on problems with more groups. We have run the same PWR-900 problem with 44 groups, resulting in 1,728,684,249,600 DOF. Four versions of the problem are executed: power iteration (PI) with GS plus GMRES, PI with GMRES, Arnoldi with GS plus GMRES, and Arnoldi with GMRES. With the multigroup GS plus GMRES option, multilevel partitioning is performed only over the upscattering region of the scattering matrix. For this option, the lower-triangular downscatter region is replicated on each set. With the multigroup GMRES option, multilevel partitioning is performed over the whole scattering matrix, regardless of its structure. Each problem was run with 10,200 blocks and 11 sets, resulting in 112,200 domains.

The results of these runs are given in terms of the weak-scaling parallel efficiency that is defined as

$$\varepsilon = \frac{\tau_{\mathrm{ref}}}{\tau_{\mathrm{P}}}\left(\frac{\mathrm{DOF_P}}{\mathrm{DOF_{ref}}}\right), \quad \tau = t \times N_p \,. \tag{26}$$

Here, t is the wall-clock time and N_p is the number of processors (cores). The "ref" subscript denotes a reference problem run, and the "P" subscript refers to the target problem. Weak-scaling curves for the four multilevel solvers are shown using the PI/GS plus TTG problem as the reference in Figure 6. These results show that (1) the multilevel parallel decomposition allows scaling up to $O(100K)$ cores and (2) the solvers combined with the multilevel parallel decomposition are generally more efficient than

standard PI/GS-based schemes (even with acceleration). In particular, the Arnoldi eigenvalue solver is significantly more efficient than PI, and we expect this efficiency difference to become larger as tighter eigenvalue tolerances are required. In general, reactor design calculations require eigenvalue tolerances of approximately 1×10^{-5}, 2 orders of magnitude tighter than our current 1×10^{-3} tolerance. Early test problems indicate that the efficiency of the Arnoldi solver approaches 6 times the efficiency of PI at these tighter tolerances.



(a) (b)

Figure 6. (a) Weak scaling efficiencies for the 44-group version of the PWR-900 benchmark, and (b) peak machine efficiencies on Jaguar XT5.

7. Conclusions

We have implemented a new suite of eigenvalue and multigroup solvers in the Denovo radiation transport code that utilize a multilevel energy decomposition. Using these solvers in concert with the multilevel parallel decomposition allows Denovo to scale to $O(100K)$ processors on leadership-class hardware. Original space-angle parallel methods were fundamentally limited in how much machine resource could be used for a given problem size. The multilevel decomposition overcomes this barrier and should allow scaling to exascale-level platforms and beyond.

Acknowledgments

The authors wish to thank Kevin Clarno and Brenden Mervin, ORNL, for help in generating the multigroup cross-section sets. Work for this paper was supported by Oak Ridge National Laboratory, which is managed and operated by UT-Batelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] Clarno K. 2007. ORNL LDRD Report. Tech. Rep. D06–087 Oak Ridge National Laboratory.
[2] Zhong Z, Downar T, Xu Y, Williams M, and DeHart M. 2006. *Nuclear Science and Engineering* **154**.
[3] Baker R and Koch K. 1998. *Nuclear Science and Engineering* **128** 312–320.
[4] Clarno K. 2007. *Transactions of the American Nuclear Society* **97**.
[5] Pautz S. 2002. *Nuclear Science and Engineering* **140** 111–136.
[6] Evans T, Stafford A, Slaybaugh R, and Clarno K. 2010. *Nuclear Technology* **171** 171–200.
[7] Evans T, Clarno K, and Morel J. 2010. *Nuclear Science and Engineering* **165** 292–304.

[8] Baker C, Hetmaniuk U, Lehoucq R, and Thornquist H. 2009. *ACM Transactions on Mathematical Software* **36**.
[9] Courau T. 2009. Specifications of a 3D PWR core benchmark for neutron transport. Technical Note CR-128/2009/014 EDF-SA.
[10] Oak Ridge National Laboratory. 2009. *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluations* Version 6.

Appendices

Appendix A

SUNDAY, JULY 11	
Conference Registration **Walker Room foyer**	
6:00 pm	Conference check-in open - Participants to obtain conference materials and sign up birds-of- a-feather discussion sessions - Speakers to finalize and submit electronic versions of their presentations - Session Chairs to conduct briefing session with speakers - Project discussion opportunities for SciDAC collaborators
Poster Session - Session Chair: Jeff Kuehn **Walker Room**	
7:00 - 9:00	Computational Sciences Graduate Fellows (CSGF) Poster Session

MONDAY, JULY 12 Ballroom *(unless otherwise noted)*			
7:30 - 8:30	WORKING CONTINENTAL BREAKFAST: - Project discussion opportunities for SciDAC collaborators - Session Chairs to finalize introductions - Boardroom and birds-of-a-feather session sign-ups Ballroom		
Opening Session - Chair: Thomas Zacharia, ORNL			
8:30 - 9:00	Introduction	**SciDAC Update**	**Michael Strayer** DOE SciDAC Director
9:00 - 9:45	Keynote	Computational Challenges in Science, Energy, and National Security	Undersecretary **Steven Koonin**
9:45 - 10:15	BREAK: Discussions continue, refreshments served Ballroom foyers		
Technical Session - Chair: Thomas Zacharia			
10:15 - 10:45	Exascale Simulation	Observations on Justifying the Case for Exascale Computing	**Alvin Trivelpiece**
10:45 - 11:15	Exascale Simulation	A DOE Laboratory Plan for Providing Exascale Applications and Technologies for Critical DOE Mission Needs	**Andy White**, LANL **Rick Stevens**, ANL
11:15 - 11:45	Exascale Simulation	CASL: The Consortium for Advanced Simulation of Light Water Reactors – a DOE Energy Innovation Hub	**Doug Kothe**, ORNL
11:45 -12:45	WORKING LUNCH: birds-of-a-feather roundtable discussions Broad Street		
Technical Session - Chair: Thomas Schulthess, ETH			
12:45 - 1:30	Materials	Materials for energy: can high performance computations make a difference?	**Eric Schwegler**, LLNL
1:30 - 2:00	Materials	The Materials Genome Project: Materials Design with High-Throughput ab-initio Computing	**Anubhav Jain**, MIT
2:30 - 3:00	Materials	Advancing our understanding of high-temperature superconductors through extreme scale computing	**Thomas Maier**, ORNL
3:00 - 3:30		Break: Discussions continue, refreshments served Ballroom foyers	

DUAL TRACK

Ballroom			Amphitheater		
Technical Session – Chair: Bill Gropp / Kathy Yelick			**Technical Session –Chair: Suzy Tichenor**		
3:30 - 4:15	HPC Performance	**John Mellor-Crummey**, Rice A Slice of CScADS	3:30 - 4:15	Industry	**Steven Sawyer**, Boeing Commercial Airplanes Regarding the Necessary Developments for Applying Large-Scale Computing to Commercial Aviation
4:15 - 4:45	HPC Performance	**Pat Worley**, ORNL Performance Portability and Optimization for Strong Scaling: Examples from the Community Climate System Model Pat Worley, Art Mirin, Mariana Vertenstien	4:15 - 4:45	Industry	**Michael Henderson**, BMI Corporation The Impact of the XT5 Jaguar Super Computer on Large Truck Aerodynamic Design
4:45 - 5:15	HPC Performance	**Allan Snavely**, SDSC Towards Integrated Power-Balanced Computing Allan Snavely, SDSC	4:45 - 5:15	Industry	**Kelly Anderson**, Procter & Gamble Cleaning up with HPC: Multiscale Simulations of Surface Active Molecules
Vendor Session - Chair: Buddy Bland, ORNL					
5:15 – 5:45	Break: Discussions continue Ballroom foyers				
5:45 - 6:15	Vendor	Exascale Technology Trends and Resulting End-User Challenges			**Alan Gara**, IBM
6:15 - 6:45	Vendor	Exascale Analytics			**Eng Lim Goh**, SGI
6:45 - 9:00	**Poster Session - Chair: Jeff Kuehn** **Walker Room**				
SciDAC projects					

TUESDAY, JULY 13

~ Ballroom *(unless otherwise noted)* ~

7:30 - 8:30	WORKING CONTINENTAL BREAKFAST: - Project discussion opportunities for SciDAC collaborators - Session Chairs to finalize introductions - Boardroom and birds-of-a-feather session sign-ups Ballroom foyers		
Technical Session - Chair: Tony Mezzacappa			
8:30 – 9:00	Breakthroughs	Large-Scale Simulations of Nuclear Dynamics	**Joe Carlson**, LANL
9:00 - 9:30	Breakthroughs	Hybrid Quantum Simulations of Niomolecules: the Role of Copper in Neurodegenerative Diseases (J. Bernholc, M. Hodak, W. Lu, F. Rose)	**Jerry Bernholc**, NCSU
9:30 – 10:00	Breakthroughs	Fully 3D Multiple Beam Dynamics Processes Simulation for the Fermilab Tevatron (E. Stern, J. Amundson, P. Spentzouris and A. Valishev)	**Eric Stern**, FNAL
10:00 - 10:30	BREAK: Discussions continue, refreshments served Ballroom foyers		
Technical Session - Chair: Petr Plechac / Lori Diachin			
10:30 – 11:15	Applied Math	Adaptive Computational Fluid Dynamics: Petascale and Beyond (K. Jansen, O. Sahni, A. Ovcharenko, M. Shephard, M. Zhou)	**Ken Jansen**, RPI
11:15 – 11:45	Applied Math	The Math Behind MADNESS (G. Beylkin, G. Fann, R. Harrison, J. Hill, J. Jia, M. Mohlenkamp, M. Ou, F. Perez and D. Galindo)	**George Fann**, ORNL
11:45 – 12:15	Applied Math	Numerical Simulation of Phase Space Advection in Gyrokinetic Models of Fusion Plasmas (M. R. Dorr, R. H. Cohen, P. Colella, J. A. F. Hittinger, D. F. Martin)	**Milo Dorr**, LLNL
12:15 – 1:15	WORKING LUNCH: birds-of-a-feather roundtable discussions Broad Street		

Technical Session - Chair: Jeff Nichols			
1:15 – 2:00	Chemistry	Life in the Fast Lane: scientific software in the exascale era	**Robert Harrison**, ORNL/UTK
2:00 – 2:30	Chemistry	Toward Accurate Ab Initio Calculations on Complex Molecular Processes (Y. Kholod, M. Lamm, T. Windus)	**Mark Gordon**, Iowa State
2:30 – 3:00	Chemistry	Developing Coarse-Grained Models for the Colecular Simulation of the Self Assembly of Skin Lipids (K. R. Hadley, C. McCabe)	**Clare McCabe**, Vanderbilt
3:00 – 3:30	BREAK: Discussions continue, refreshments served Ballroom foyers		

DUAL TRACK

Ballroom			Amphitheater		
Technical Session – Chair: Bill Gropp / Kathy Yelick			**Technical Session – Chair: Stephane Ethier**		
3:30 – 4:15	Computer Science	**Jeff Hollingsworth**, U. Maryland Recent Results in Auto-Tuning and Performance Tools	3:30 - 4:15	Fusion Energy	**Weixing Wang**, PPPL Non-diffusive Momentum Transport and Intrinsic Rotation in Tokamaks
4:15 - 4:45	Computer Science	**Richard Vuduc**, Georgia Tech On the Limits and Opportunities of GPU Acceleration	4:15 – 4:45	Fusion Energy	**Linda Sugiyama**, MIT Edge Instabilities and Magnetic Tangles - a New Picture of Fusion Plasmas
4:45 - 5:15	Computer Science	**Rajeev Thakur**, ANL MPI at Exascale: What will scale and what needs to change?	4:45 - 5:15	Fusion Energy	**Michael Barnes**, U. Oxford Direct Multi-Scale Coupling of a Plasma Transport Code to Gyrokinetic Turbulence Codes, with Comparison to Fusion Experiments. (M. Barnes, W. Dorland, T. Görler, G.W. Hammett, F. Jenko)
5:15 – 5:45	BREAK: Discussions continue Ballroom foyers				

Vendor Session - Chair: Buddy Bland, ORNL Ballroom			
5:45 – 6:15	Vendor	OpenCL Acceleration for HPC	**William C. Brantley**, AMD
6:15 – 6:45	Vendor	Throughput Computing: To ExaScale and Beyond	**Bill Dally**, NVIDIA
6:45 – 9:45	**Poster Session - Chair: Jeff Kuehn ~ Walker Room ~**		
SciDAC projects			

WEDNESDAY, JULY 14

7:30 - 8:30	WORKING CONTINENTAL BREAKFAST: - Project discussion opportunities for SciDAC collaborators - Session Chairs to finalize introductions - Boardroom and birds-of-a-feather session sign-ups **Ballroom**		
Technical Session - Chair: Phil Jones ~ Ballroom ~			
8:30 – 9:15	Climate	ISICLES: Ice Sheet Initiative for Climate Extremes	**Esmond Ng**, LBNL
9:15 – 9:45	Climate	Using single-forcing factor experiments to explore climate-biogeochemistry feedbacks in CESM1 (P. Thornton, K. Lindsay, M. Branstetter, and the CCSM Biogeochemistry Working Group)	**Peter Thornton**, ORNL
9:45 – 10:15	Climate	Model for Prediction Across Scales: A Multi-Resolution Approach to Climate Modeling (T. Ringler, B. Skamarock, M. Gunzburger)	**Todd Ringler**, LANL
10:15 – 10:45		BREAK: Discussions continue, refreshments served, Ballroom foyers	

DUAL TRACK

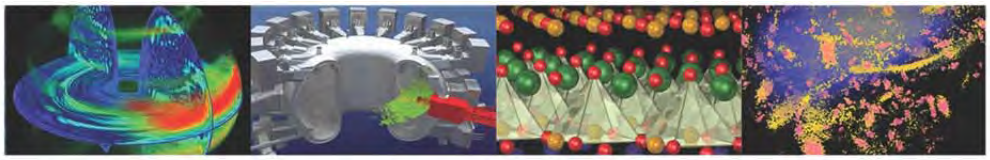
Ballroom			Amphitheater		
Technical Session - Chair: Jackie Chen			**Technical Session - Chair: Richard Brower**		
10:45 – 11:30	Combustion	**Bob Moser**, UT Austin Two Opportunities in Extreme Computing: Turbulence Simulation and Uncertainty Quantification	10:45 – 11:30	Lattice QCD	**Paul Mackenzie**, FNAL Lattice QCD and the Standard Model in the Exascale Era
11:30 – 12:00	Combustion	**Diego A. Donzis**, Texas A&M University Toward Petascale Simulations of Turbulence and Turbulent Mixing: Recent Results and Future Opportunities	11:30 – 12:00	Lattice QCD	**Julius Kuti**, UCSD Lattice GPU Computing and the Large Hadron Collider
12:00 – 12:30	Combustion	**Ray Grout,** NREL High Fidelity Simulations Towards Understanding Combustion of Evolving Fuels R.W. Grout, C.S. Yoo, E.S. Richardson, S. Hammond, J.H. Chen	12:00 – 12:30	Lattice QCD	**Peter Petreczky**, BNL Properties of Hot Strongly Interacting Matter from Large Scale Numerical Simulations
12:30 – 1:15	WORKING LUNCH: birds-of-a-feather roundtable discussions Broad Street				
Technical Session – Chair: John Blondin			**Technical Session – Chair: Sean Ahern**		
1:15 – 1:45	Astrophysics	**Alex Szalay**, Johns Hopkins University Amdahl's Laws and Extreme Data Intensive Computing	1:15 – 1:45	Visualization	**Peter Lindstrom**, LLNL Challenges of Multiplicity in Extreme-Scale Visualization
1:45 – 2:15	Astrophysics	**Christian Cardall**, ORNL / UT Towards Exascale Supernova Simulations with GenASiS (C. Cardall, E. Endeve, R. Budiardja, P. Marronetti, A. Mezzacappa)	1:45 – 2:15	Visualization	**Valerio Pascucci**, Utah Topology Rich Visualization for Interactive Feature Definition and Exploration (V. Pascucci, A.Gyulassy)
2:15 – 2:45	Astrophysics	**Alan Calder**, Stony Brook University Evaluating Systematic Dependencies of Type Ia Supernovae. (A.P. Jackson, B.K. Krueger, D.M. Townsley, E.F. Brown, F.X. Timmes, D.A. Chamulak)	2:15 – 2:45	Visualization	**John Owens**, UC Davis What's New with the GPU? GPUs for Scientific Computing and Visualization
2:45 – 3:15	BREAK: Discussions continued, refreshments served Ballroom foyers				
Technical Session – Chair: Peter Lichtner			**Technical Session – Chair: Panagiotis Spentzouris**		
3:15 – 4:00	Geosciences	**Gary Pope**, UT Austin Overview of CO_2 Geologic Storage Research Activities of the Center for Frontiers for Subsurface Security	3:15 – 4:00	Accelerators	**Warren Mori**, UCLA Dream Beams: Extreme Scale Computing Enabling New Accelerator Technologies for the Energy and Intensity Frontiers
4:00 – 4:30	Geosciences	**Hajime Yamamoto**, Taisei Corporation Massively Parallel Simulation of CO_2 Geologic Storage – An Approach for Investigating Impact on Basin-Scale Groundwater Environment	4:00 – 4:30	Accelerators	**Rich Lee**, SLAC Wakefield Computations at Extreme Scale for Ultra-short Bunches using Parallel hp Refinement (L-Q Lee, A. Candel, C. Ng, A. Kabel, L. Xiao, G. Schussmam, K. Ko)
4:30 – 5:00	Geosciences	**Glenn Hammond**, PNNL Cleaning Up the Cold War: Simulating Uranium Migration at the Hanford 300 Area (G. Hammond and P. Lichtner)	4:30 – 5:00	Accelerators	**James Amundson**, FNL Advances in large-scale computations of multi-particle beam dynamics (J. Amundson, A. Macridin, P. Spentzouris, E. Stern)

Vendor Session - Chair: Buddy Bland Ballroom			
5:30 – 6:00	Vendor	**Steve Scott** Meeting the Exascale Challenges	CRAY
6:00 – 6:30	Vendor	**Stephen Pawlowski** Exascale: The Next Frontier in High-Performance Computing	INTEL

Vis Night - Chattanooga Aquarium River Journey ASCR visualization competition Board buses at Market and 12th Street Charter bus will pick up outside hotel lobby every 15 minutes to / from The Chattanooga Aquarium.	
7:00 - 10:00	Scientific visualization displays and OASCR balloting Attention: Vis Night at the Aquarium does not open until 7:00 p.m. Ballots for OASCR awards will be available upon entrance to The Chattanooga Aquarium beginning at 7:00 p.m. Ballots will be collected until 8:30 p.m. Winners will be announced in the Aquarium Amphitheater after tallying.

THURSDAY, JULY 15 ~ Ballroom *(unless otherwise noted)* ~			
7:00 - 8:00	WORKING CONTINENTAL BREAKFAST: - Project discussion opportunities for SciDAC collaborators - Session Chairs to finalize introductions - Boardroom and birds-of-a-feather session sign-ups Ballroom		
Technical Session - Chair: Jeremy Smith			
8:00 - 8:35	Biology	Biomolecular Simulation – A Field Coming of Age	**Tamar Schlick**, NYU
8:35 - 9:10	Biology	Systematic and Scalable Multiscale Simulation of Multiprotein Complexes	**Greg Voth**, ORNL/UT
9:10 - 9:45	Biology	Molecular Simulation in the Energy Biosciences	**Jeremy Smith**, ORNL/UT
9:45 – 10:15	BREAK: Discussions continue, refreshments served Ballroom foyers		
Technical Session - Chairs: Rob Ross / Terrence Critchlow			
10:15 – 11:00	Data Management	Scientific Data Management Challenges and Approaches in the Extreme Scale Era	**Arie Shoshani**, LBNL
11:00 - 11:30	Data Management	From Grid to Cloud, the STAR experience	**Jerome Lauret**, BNL
11:30 - 12:00	Data Management	Mining Climate and Ecosystem Data: Challenges, Opportunities, and Some Early Results	**Vipin Kumar,** U. Minnesota
12:00 - 1:00	WORKING LUNCH: birds-of-a-feather roundtable discussions Broad Street		
Technical Session - Chair: John Turner			
1:00 - 1:30	Nuclear Energy	Three-Dimensional Full-Core Power Calculations for Pressurized Water Reactors (T. Evans, G. Davidson, R. Slaybaugh, K. Clarno)	**Tom Evans**, ORNL
1:30 - 2:00	Nuclear Energy	Computational Approach to Multiscale Multiphysics Modeling of Nuclear Reactors	**Mike Podowski**, RPI
2:00 - 2:30	Nuclear Energy	Reactors Multiscale Investigation of Irradiation Effects in Fuel, Clad and Structural Materials	**Brian Wirth**, UC Berkeley
2:30 - 3:00	BREAK: Discussions continue, refreshments served Ballroom foyers		
Technical Session - Chair: Patricia Kovatch			
3:00 - 3:30	Facilities	Bringing Users along the Road to Billion Way Concurrency	**Kethy Yelick**, NERSC
3:30 - 4:00	Facilities	The Software of the Future on the Hardware of the future" (J. Hammond, W. Scullin, R. Bair)	**Jeff Hammond**, ANL
4:00 - 4:30	Facilities	GPU accelerators: The good, bad and ugly! (R. Kendall, W. Joubert, I. Carpenter, A. Tharrington, R. Sankaran, M. Eisenbach, O. Hernandez, B. Messer, R. Hartman-Baker, C. Baker, J. Turner, D. Kothe, R. Harrison, J. Larkin)	**Ricky Kendall**, ORNL
Closing Session			
4:30 - 5:00	Closing Remarks		**Thomas Zacharia**, ORNL **Lori Diachin**, LBNL
5:00	**ADJOURN**		

Appendix B



SciDAC 2010

Electronic Visualization and Poster Night

July 14
Chattanooga
Tennessee

U.S. DEPARTMENT OF ENERGY
Office of Science

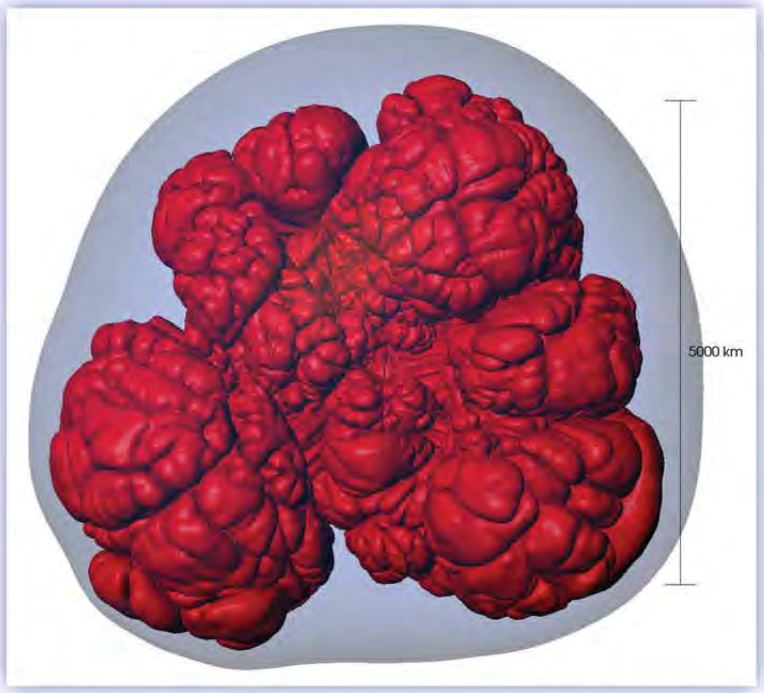
Scientific Discovery through Advanced Computing

Visualization posters are at http://computing.ornl.gov/workshops/scidac2010/vis_posters.pdf.
Visualization night competition winners are at http://computing.ornl.gov/workshops/scidac2010/viz_night_winners.pdf.

Type Ia Supernova: Turbulent Combustion on the Grandest Scale

1

Hank Childs, Haitao Ma, Stan Woosley, John Bell, Ann Almgren, Andy Nonaka



This image shows the star surface (light blue) and the flame front (red) inside of a Type Ia supernova simulation.

Video description:

Deep inside a dying star in a galaxy far, far away, a carbon fusion flame ignites. Ignition may happen in the middle or displaced slightly to one side, but this simulation explores the consequences of central ignition. In a localized hot spot, represented here by a deformed sphere with an average radius of 100 km, carbon is assumed to have already fused to iron, producing hot ash (~ten billion K) with a density about 20% less than its surroundings. As the burning progresses, this hot buoyant ash rises up and interacts with cold fuel. Rayleigh-Taylor fingers give rise to shear and turbulence, which interacts with the flame, causing it to move faster. In about two seconds, the energy released blows the entire white dwarf star up, leaving nothing behind but a rapidly expanding cloud of radioactive nickel, iron, and other heavy elements. A Type Ia supernova is born.

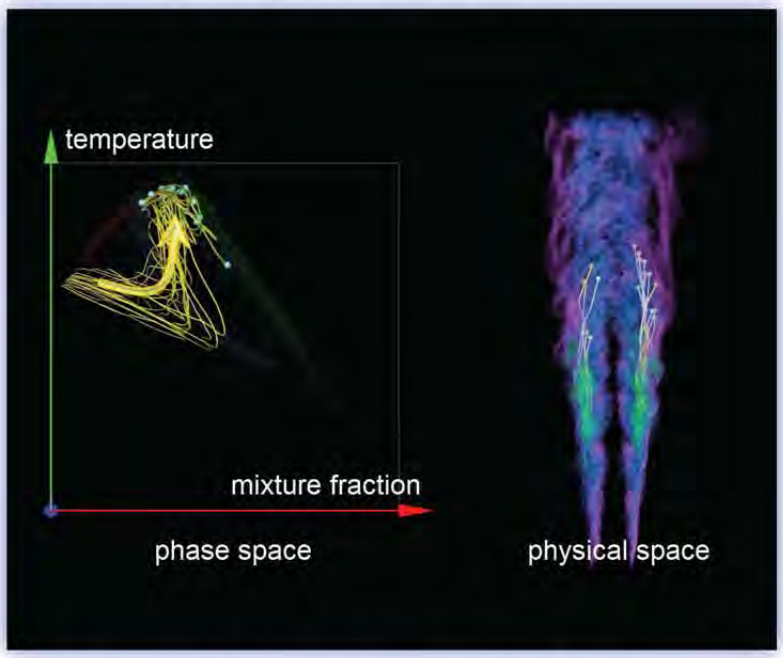
SciDAC 2010

Dual-Space Visual Analysis of a Direct Numerical Simulation of Lifted Turbulent Jet Flames

2

Simulation
Chun Sang Yoo, Edward Richardson, Ramanan Sankaran, Jacqueline H. Chen

Visualization
Hongfeng Yu, Jishang Wei, Ray W. Grout, Kwan-Liu Ma



Dual-Space Visual Analysis of a Direct Numerical Simulation of Lifted Turbulent Jet Flames

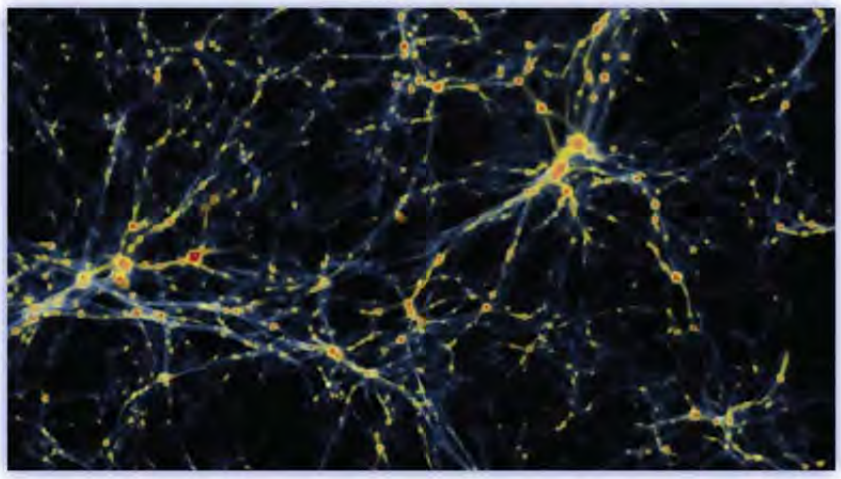
Video description:
Recent combustion simulations of a turbulent lifted autoignitive ethylene/air jet flame were instrumented with particles originating in both the fuel and oxidizer streams. The particle trajectories provide lagrangian statistics correlating temperature and mixture fraction, two key combustion parameters, encapsulating the unsteadiness associated with turbulent mixing and autoignition. The following visualization shows time-varying particles and how they are clustered into groups with distinct characteristics. The relationship between the particle trajectories in phase and physical space is highlighted by the linked images and provides combustion scientists with detailed information regarding the evolution of fluid parcels traversing a turbulent autoignitive environment.

SciDAC 2010

Searching for Baryon Acoustic Oscillations in Intergalactic Absorption, The Expanding Universe 3

Science
Michael L. Norman, Robert Harkness, Pascal Paschos, Rick Wagner

Visualization
Mark Hereld, Joseph A. Insley, Eric C. Olson, Michael E. Papka



This figure shows the density field of the baryon acoustic oscillations simulation, illustrating the gravitational clumping that occurs even as the volume continues to expand.

Video description:

This simulation follows the growth of density perturbations in both gas and dark matter components in a volume 1 billion light years on a side beginning shortly after the Big Bang and evolved to half the present age of the universe. It calculates the gravitational clumping of intergalactic gas and dark matter modeled using a computational grid of 64 billion cells and 64 billion dark matter particles. The simulation uses a computational grid of 4096^3 cells and took over 4,000,000 CPU hours to complete.

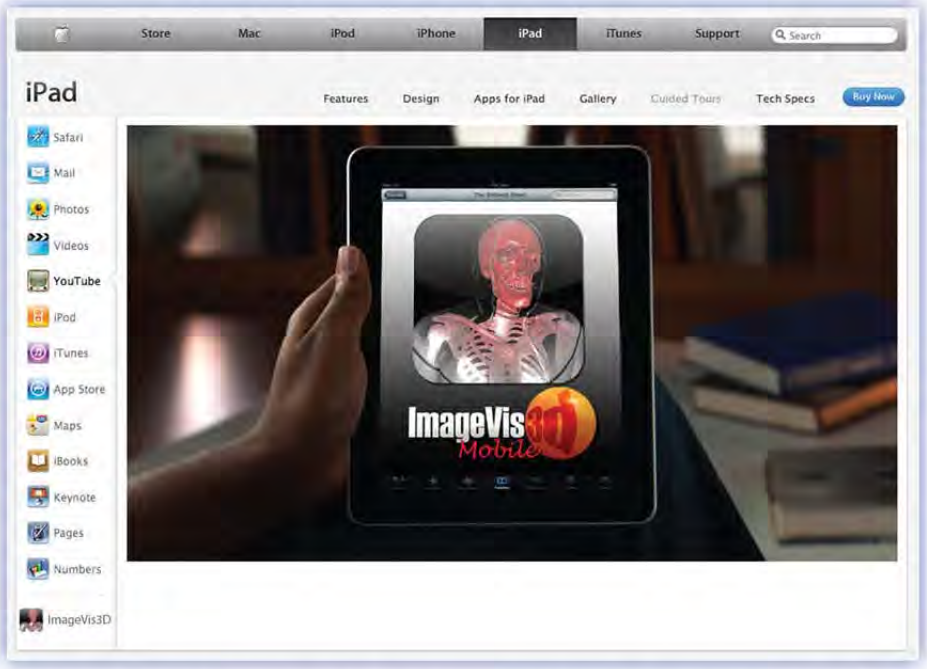
The visualization shows the density value, whose range spans over 6 orders of magnitude. The animation illustrates the expansion of the universe over time, and highlights how individual structures (such as galaxy clusters) collapse due to gravity, while simultaneously being pushed further and further from each other. The animation was made from 148 terabytes of data generated on Kraken at the National Institute for Computational Sciences and transferred to Argonne over ESnet. The rendering was performed on Eureka at the Argonne Leadership Computing Facility.

SciDAC 2010

iMageVis3D Mobile

4

Tom Fogal, Jens Krueger, Chems Touati



ImageVis3D mobile is a mobile volume rendering application that allows one to visualize and share your data while you are on the go.

Video description:

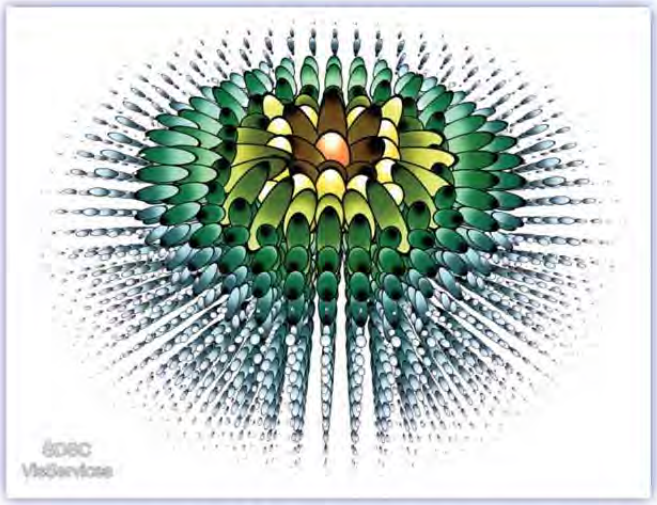
While mobile visualization is a popular field, due to the high computational cost there has been relatively little work investigating volume rendering on mobile devices. ImageVis3D mobile succeeds in providing interactive renderings which clearly detail the features of a data set. The tool makes it easy for you and your colleagues to set up collaboration-specific networks to distribute and visualize data with ease.

SciDAC 2010

GlyphSea

5

Amit Chourasia, Emmett Mcquinn, Bernard Minster, Jurgen Schulze, Geoffrey Ely, Kim Olsen, Yifeng Cui, Alexei Kritsuk, Sergey Ustyugov, Mike Norman, Kim Olsen, Steve Day, Thomas Jordan



The figure shows velocity field for a point source earthquake simulation where the size and color of ellipsoidal glyph indicate velocity magnitude. Vector orientation is encoded as a white and a black spot on the pole of the ellipsoid, where the white spot indicates the vector direction towards the viewer and black spot away from the viewer.

Video description:

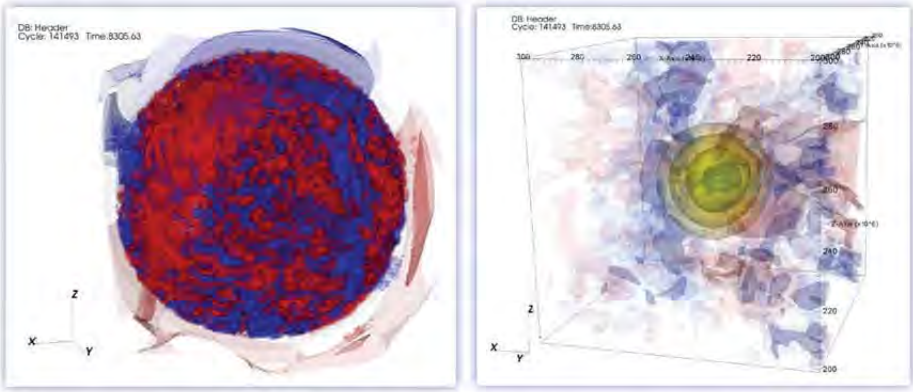
Vector visualization is an important area to investigate scientific data in many domains. Prior visualization strategies have primarily focused on maps and volumetric rendering of scalar fields, which do not permit a full characterization of underlying phenomena. We have developed an interactive glyph visualization application that allows scientists to explore vector fields. We implement novel technique of procedural dipole and cross mark texturing to encode and display vector data. We employ a novel lattice method to show neighborhood, which also enables to distinguish glyphs. We further enhance visualization by using screen space ambient occlusion, jitter, outline halos and displacement. By combining contextual information with glyphs, such as geographic maps and representations of crustal structure with three-dimensional isosurfaces, scientists are able examine vector fields in a rich and interactive multivariate environment. We demonstrate the results for time-dependent vector data from three different simulations.

SciDAC 2010

The Convective Phase Preceding Type Ia Supernovae Explosions

6

M. Zingale, A. S. Almgren, J. B. Bell, C. M. Malone, A. Nonaka, G. H. Weber, S. E. Woosley



MAESTRO simulation of the final hours of convection in a white dwarf preceding the explosion of a Type Ia supernova. The movie shows the radial velocity (red and blue contours) and the nuclear energy generation (yellow/green/purple contours). The right panel shows the inner 1000 km^3.

Video description:

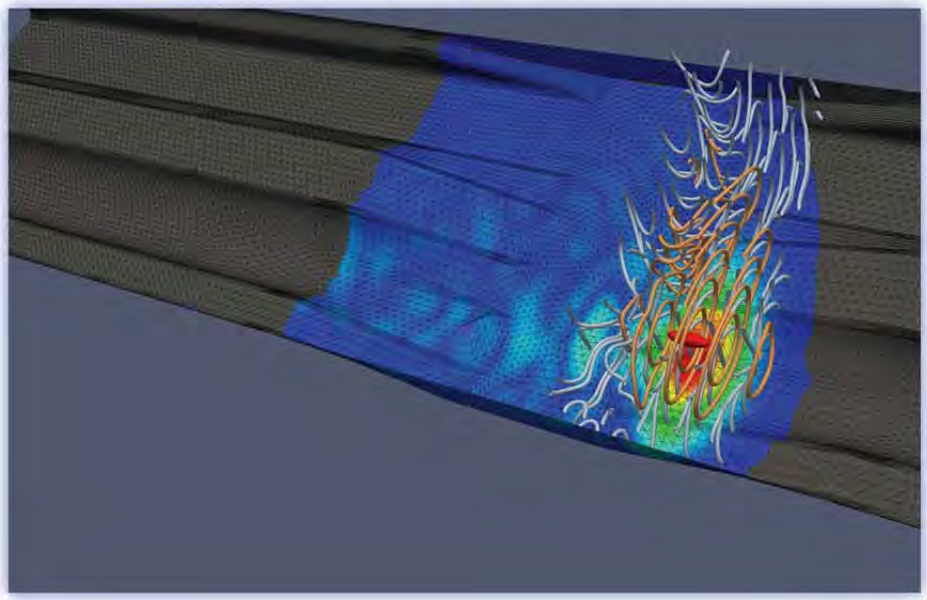
Type Ia supernovae are believed to result from the thermonuclear explosion of a Chandrasekhar mass white dwarf. For centuries preceding the ignition, carbon fusion reactions at the center drive convection throughout the star. The details of this convection are important to understanding the subsequent explosion. We used the low Mach number hydrodynamics code, MAESTRO, to model the last hours of convection preceding the ignition. During this convective phase, buoyant plumes rise outward from the center, cooling via expansion. The simulation ends when this expansion can no longer quench the burning and a flame front is born (ignition). The low Mach number hydrodynamics algorithm was critical to performing this simulation, as the flow is highly subsonic, putting it out of reach of traditional hydrodynamics codes.

SciDAC 2010

Electromagnetic Wake in an Energy Recovery Linac Vacuum Chamber with Moving Simulation Window

7

Greg Schussman, Rich Lee, Liling Xiao, Cho-Kuen Ng, Ken Moreland



Electromagnetic wakefields from simulation of a short electron bunch in a Vacuum Chamber of the Energy Recovery Linear Accelerator.

Video description:

This animation shows electromagnetic wakefields from a short electron bunch traveling through a Vacuum Chamber of the Energy Recovery Linac. Short bunches require such a fine mesh for accuracy that simulation size becomes excessive. In this simulation, computation was limited to only the volume within a moving window surrounding the bunch. When the bunch has moved sufficiently far, the window jumps forward.

An oversized red pill represents the 0.3 mm bunch. On the surface pseudo-coloring shows the electric field strength, with uncomputed values (outside the moving window) or zero values both shown as grey and low through high values shown as rainbow colors, blue through red. Only the coarse mesh is shown because this resolution is insufficient for extremely fine details. Electric field lines are shown in pale blue. Magnetic field lines form loops and are shown in light orange. This animation was rendered in ParaView.

SciDAC 2010

Verification Study of Buoyancy-Driven Turbulent Nuclear Combustion for Three Different Physical Situations

8

Ray Bair, Anshu Dubey, Robert Fishe, Jonathan Gallagher, Randy Hudson, Don Lamb, Dongwook Lee, John Norris, Mike Papka, Katherine Riley, Dean Townsley

Understanding how Buoyancy-driven turbulent nuclear combustion occurs in different physical situations is a key component to understanding the energy dynamics of the deflagration phase of Type Ia supernovae.

Video desription:

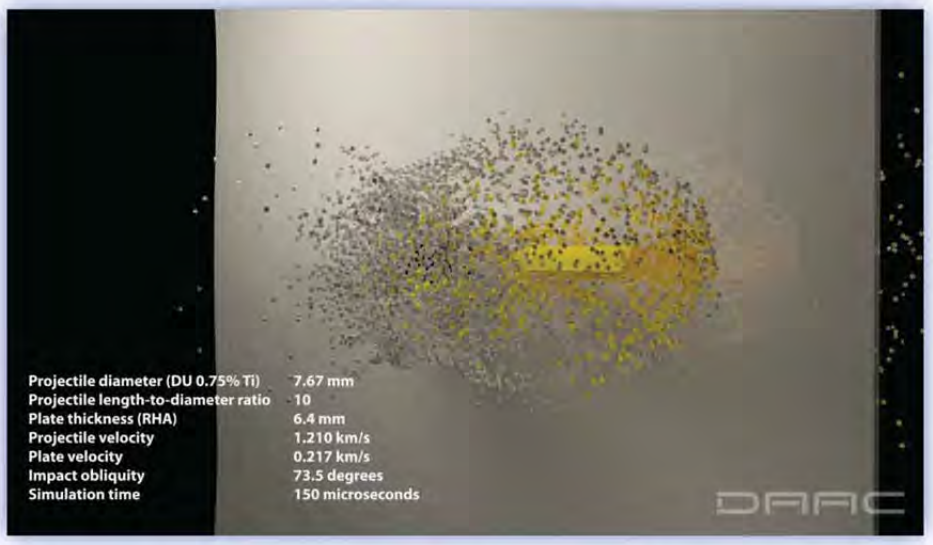
Buoyancy-driven turbulent nuclear combustion determines the rate of nuclear burning during the deflagration phase (i.e., the ordinary nuclear flame phase) of Type Ia supernovae, and hence the amount of nuclear energy released during this phase. It therefore determines the amount the white dwarf star expands prior to initiation of a detonation wave, and so the amount of radioactive nickel and thus the peak luminosity of the explosion. However, this key physical process is not fully understood. To better understand this process, the Flash Center has conducted an extensive series of large-scale 3D simulations of buoyancy-driven turbulent nuclear combustion for three different physical situations. This movie shows the results for some of these simulations.

SciDAC 2010

Hybrid Particle Finite Element Simulation of Impact, Perforation, and Fragmentation

9

Randall Hand, Miguel Valenciano, Eric Fahrenthold, Kwon Joong Son



A 7.67mm rod projectile impacting a 6.4mm plate at 1.21 km/s.

Video description:

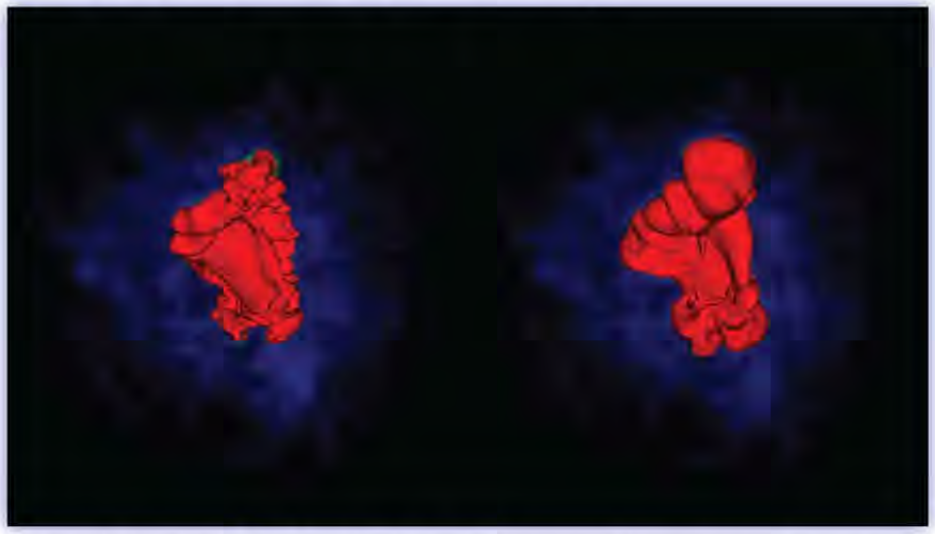
The 2 simulations model impact experiments conducted at the University of Dayton and Los Alamos National Laboratory. The simulations employ a hybrid particle finite-element method and the parallel code EXOS, both developed at the University of Texas at Austin.

SciDAC 2010

Turbulence-Flame Interaction of Initial Sparks of a Type Ia Supernova

10

Alan C. Calder, Aaron P. Jackson, Dean M. Townsley



The evolution of the first sparks (red) of a thermonuclear supernova interacting with a turbulent velocity field (blue). A model of the turbulence-flame interaction (TFI) utilizing a local measure of the turbulent velocity was developed (right) and compared to an indirect treatment of the TFI using only buoyancy effects (left). We find that our local treatment of TFI produces a larger enhancement to the flame speed as indicated by a larger volume enclosed by the flame surfac.

Video description:

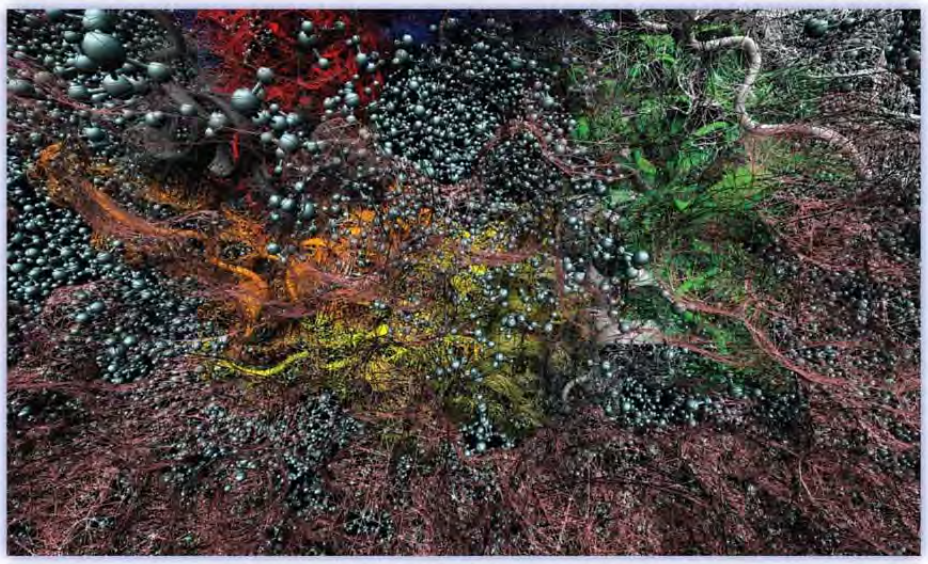
Type Ia supernovae are bright stellar explosions thought to occur when a runaway thermonuclear reaction incinerates a compact star known as white dwarf. In many models, the explosion begins with a flame born in the turbulent environment near the center of the white dwarf. The effect of turbulence on the evolution of the nascent flame is incompletely understood and is the subject of active research. We present two simulations of the early flame evolution in a supernova. One incorporates a physically-motivated turbulence-flame interaction model that utilizes a local measure of the turbulence, while the other relies only on indirect buoyancy effects. In the renderings, the red isosurface represents the flame front and the blue volume rendered features the magnitude of the vorticity of the turbulent flow. The enhanced flame speed of the local interaction may be observed as the increased volume of burned material enclosed by the isosurface.

SciDAC 2010

Computational Study of the Molecular Mechanism of Alzheimer's Disease

11

Igor F. Tsigelny, Yuriy Sharikov, Eliezer Masliah



Video description:

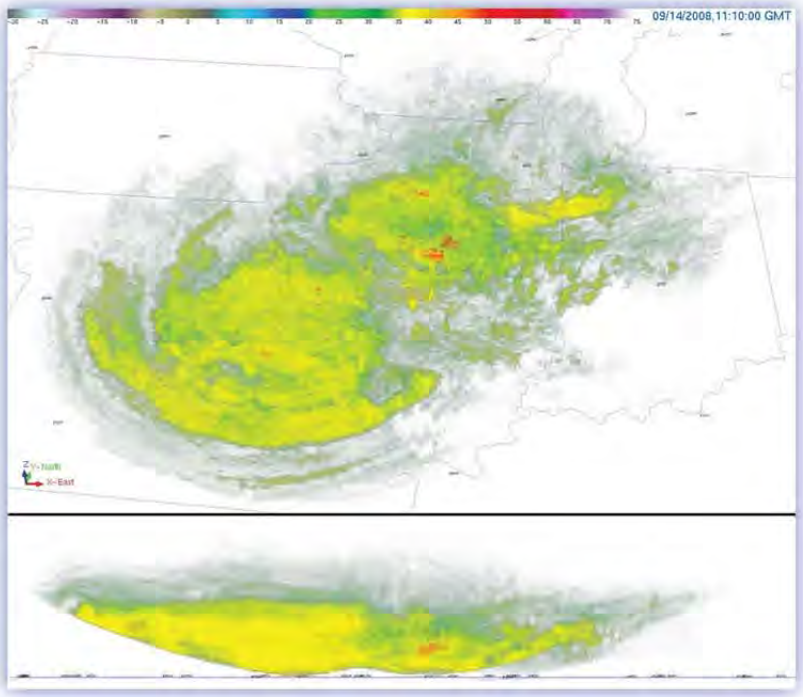
In our study we conducted all atom molecular dynamic simulations of the full-length amyloid beta 1-42 (Aβ) peptide, then studied possible annular oligomer aggregates formed by different typical conformers of Aβ. The various annular aggregates are possible to be formed. Our study of the aggregates having the best energy contacts between themselves and with the membrane brought a limited number of them for further study. These aggregates were situated on the membrane and molecular dynamic simulations of the system have been conducted. The provided images show the possible penetration of the Aβ aggregates to the membrane with the membrane perforation. Such pores can lead to uncontrolled influx of various ions to the neuronsl, including Calcium 2+ ions, that can trigger the signaling cascade of cell death. The damage and death of these neurons is associated with the Alzheimer's disease.

SciDAC 2010

Large-Scale Volumetric Visualization of Doppler Reflectivity Data

12

Peter Kristof, Jin Ryong Kim, Bedrich Benes, Lan Zhao, Carol X. Song



Volumetric visualization of Hurricane Ike, which occurred in Indiana and Illinois states on September 14, 2008, with visualization importance given to medium to heavy rainfall regions.

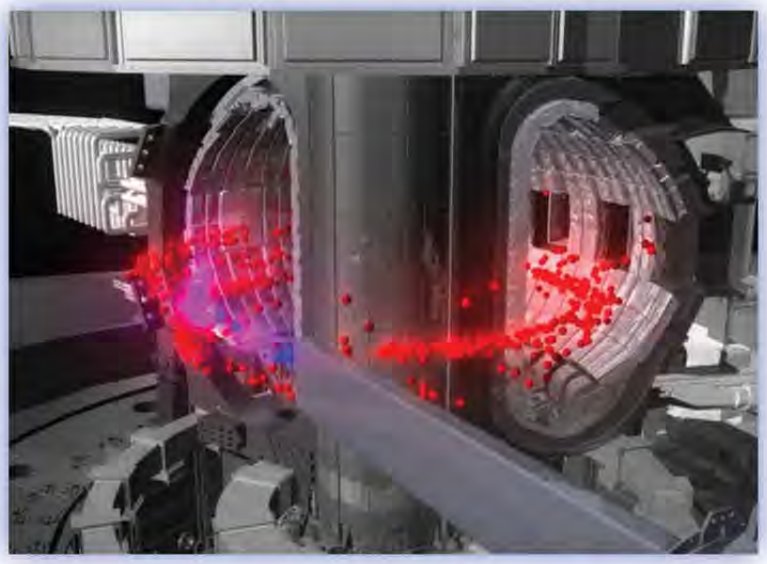
Video description:

The super resolution NEXRAD (Next-Generation Radar) Level II data, aka Weather Surveillance Doppler Radar 1988 (WSR-88D) data, provides critical information on reflectivity, wind velocity and spectrum width for the entire United States. The goal of this work is to develop a framework that enables multiple users to interactively access, analyze and visualize the Doppler reflectivity data in 3D to study near real-time weather events. To provide interactive high-quality volumetric weather visualization, we combined two approaches dealing with large-scale storage of global weather data and out-of-core volume rendering using CUDA ray casting. The results of our work show that the reflectivity data from multiple radars can be preprocessed into a data format that is efficient for large-scale volumetric visualization in near-real time and requires minimal run-time processing.

SciDAC 2010

Clean Energy for the Future with the ITER Reactor 13

Jamison Daniel, David Pugmire, Michael Matheson, Sean Ahern



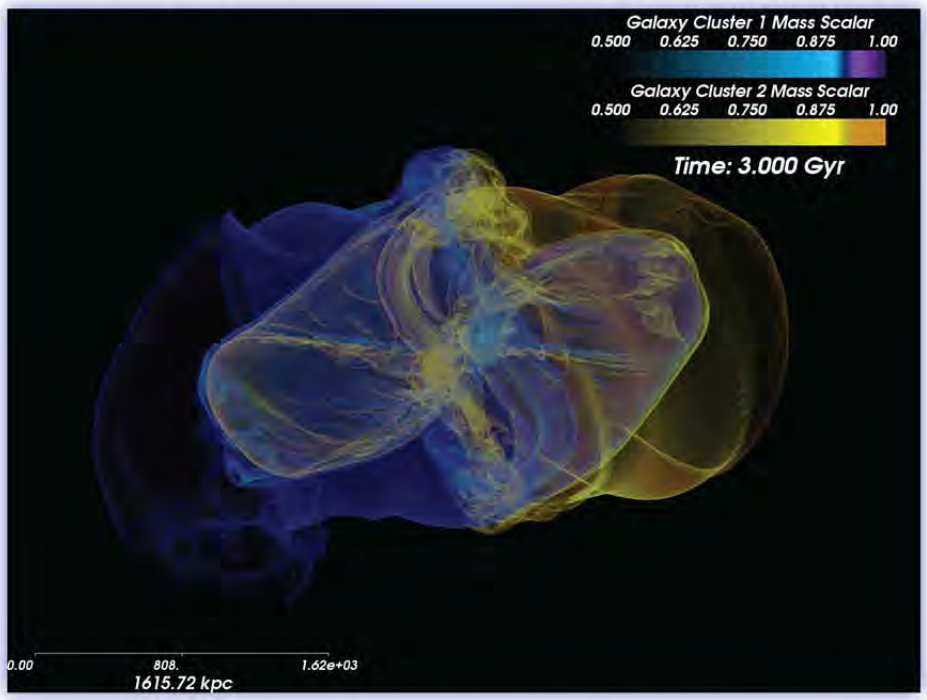
Fusion energy production by alpha particle injection.

Video description:
Within the tokamak, we see field coils, the vacuum vessel, blanket modules, divertor cassettes, and equipment for heating and diagnostics. Visible is one of the high-energy neutral beam injectors that heat and drive the plasma. The fusion fuel consists of deuterium and tritium, which are isotopes of ordinary hydrogen (shown in red and green in the visualization). To produce fusion reactions, the fuel must be heated to a temperature of about one hundred million degrees - about ten times the temperature of the core of the sun. At such temperatures, the electrons of atoms are stripped from the nuclei forming as a state of matter called plasma. One method of achieving these high temperatures is by injecting beams of high-energy neutral atoms into the tokamak. An initial plasma is formed and heated by driving an electric current through the fuel gas in the tokamak chamber. When the plasma reaches a sufficient density and temperature, the injectors are turned on. Because injected atoms are electrically neutral, they are unaffected by the magnetic field and can penetrate deep into the plasma before being ionized by collisions with plasma particles. These very energetic beam ions are trapped by the magnetic field and circulate throughout the plasma, colliding with the plasma particles and transferring energy to them. As the temperature of the plasma rises, due to the beam heating, fusion reactions between the plasma deuterium and tritium begin to occur.

SciDAC 2010

Binary Galaxy Cluster Merger, Simulated using the Flash Code, Mass Ratio 1:1, with an Offset Impact, 4 different views 14

John Zuhone, Don Lamb



Flash simulation of a Galaxy Cluster merger with a mass ratio of 1:1, and an offset impact. The visualization shows how Dark Matter (visualized as particles) and ordinary matter interact in the merger event.

Video description:

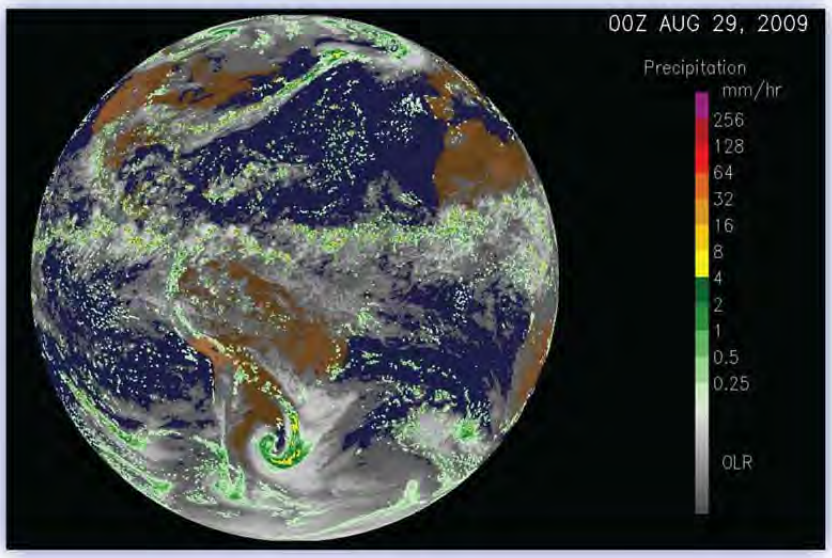
Since structure in the universe forms in a bottom-up fashion, with smaller structures merging to form larger ones, modeling the merging process in detail is crucial to our understanding of cosmology. At the current epoch, we observe clusters of galaxies undergoing mergers. It is seen that the two major components of galaxy clusters, the hot intra-cluster gas and the dark matter, behave very differently during the course of a merger. Using the N-body and hydrodynamics capabilities in the FLASH code, we have simulated a suite of representative galaxy cluster mergers, including the dynamics of both the dark matter, which is collisionless, and the gas, which has the properties of a fluid. 3-D visualizations such as these demonstrate clearly the different behavior of these two components over time.

SciDAC 2010

Dedicated High-End Computing to Revolutionize Climate Modeling: An International Collaboration

15

B. Doty, D. Achutavarier, J. Adams, E. Altshuler, P. Andrews, B. Cash, B. Huang, E. Jin, J. Kinter, L. Marx, J. Manganello, M. Satoh, C. Stan, D. Straus, H. Tomita



The image is full-disk view of the simulated Earth's atmosphere centered over the Atlantic Ocean. The image depicts the cloudiness (shades of white; based on model-simulated outgoing long-wave radiative flux) and precipitation (colors) at one instant in a long climate simulation of the summer of 2009. Because the model resolution is 7 km, nearly the entire range of multi-scale features in the cloud and rainfall fields can be discerned.

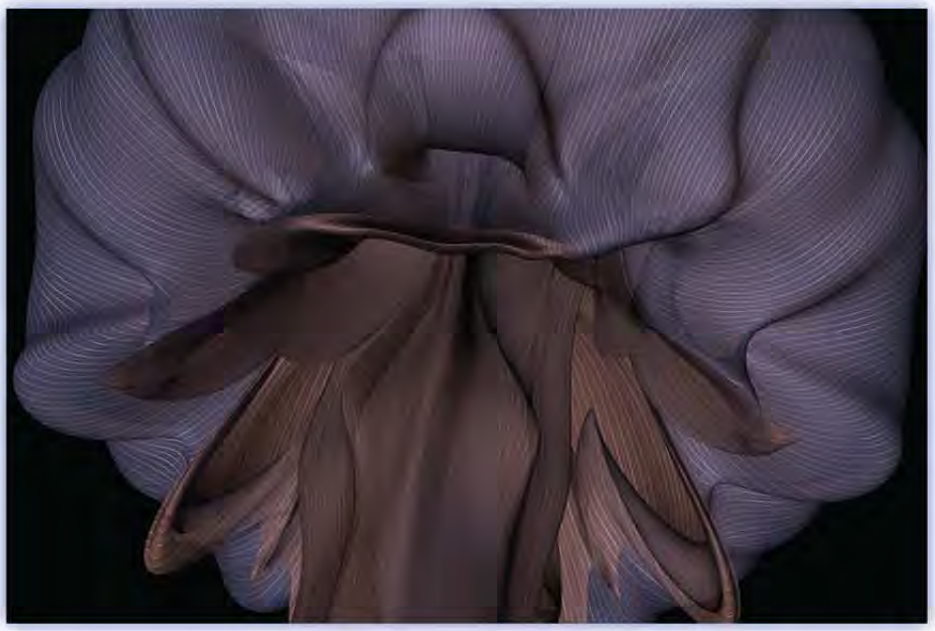
Video description:

The animation depicts a simulation of the Earth's atmosphere, produced from output of a very high-resolution model of the Earth's atmosphere, namely the Non-hydrostatic ICosahedral Atmospheric Model, developed by the University of Tokyo and the Japan Agency for Marine-Earth Science and Technology. The simulation was made as part of a collaboration involving over 30 people, from six institutions on three continents, including climate and weather scientists and modelers, and experts in high-performance computing (HPC), to demonstrate the feasibility of using dedicated resources to rapidly accelerate progress in global climate variability and change. The scientific basis for undertaking this project was established in the World Modeling Summit, held in May 2008 in Reading, UK, where there was a call for a revolution in high-resolution weather and climate modeling. The project made use of the entire 18,048-core Athena Cray XT-4 supercomputer at the University of Tennessee's National Institute for Computational Sciences (NICS).

SciDAC 2010

Visualization of Convective Flow with Integral Surfaces 16

Christoph Garth, Hari Krishnan, Kenneth I. Joy



View from below of an integral surface that visualizes a plume-like structure appearing in a convective flow.

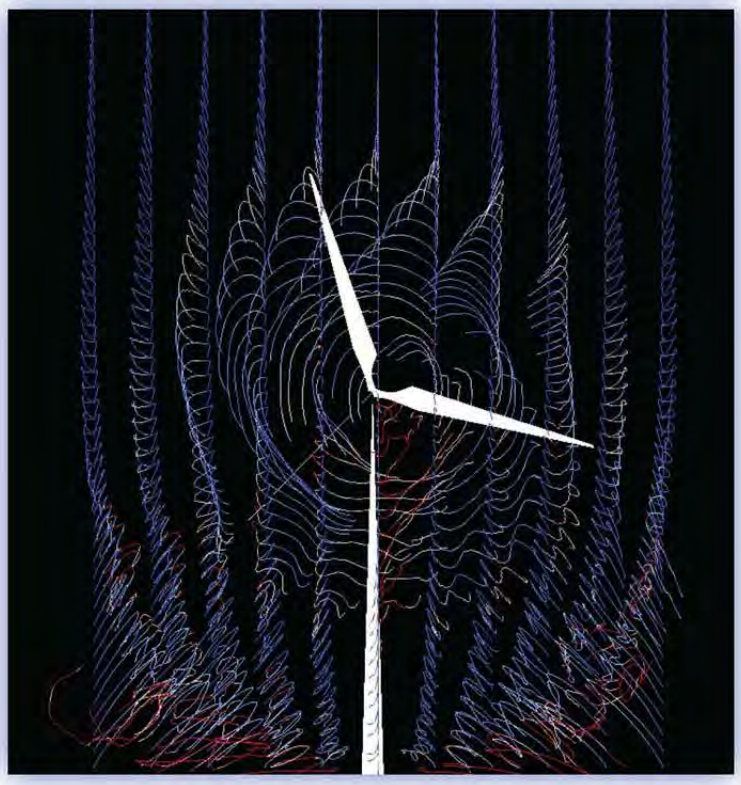
Video description:

Integral surfaces are a tool for the visualization of complex flow structures in both steady and unsteady vector fields. These surfaces are derived from the trajectories of massless particles transported by the flow. Depending on how the particles are seeded, different surface types such as path, time, and streak surfaces are obtained. We have developed algorithms to compute such surfaces in a robust and efficient manner, to allow their application to large and complex vector field data. The complex nature of integral surfaces, which can exhibit strong twisting, folding, and shearing, can be problematic in generating a graphical representation that adequately conveys the spatial and temporal characteristics of the flow under consideration. We provide several default approaches to this problem that cover a wide range of integral surface visualization scenarios.

SciDAC 2010

Visualization of a Wind Blade in a Wind Field 17

Pat Fasel, Jim Ahrens, Jon Woodring, Eunmo Koo, Rod Linn, Judy Wintercamp



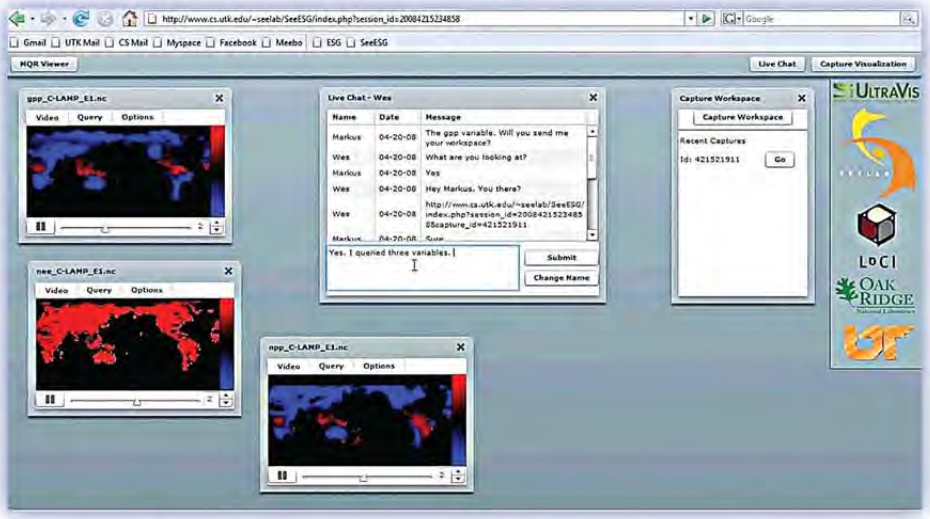
A head on view of a simulated wind blade turning in a wind field.

Video description:
Wind energy is an important key to sustainable, alternative energy sources. In wind farms, the blades turning from a wind source have a non-zero effect on the wind field. Thus, the effects of a wind blade may have impact on the efficiency of other blades in a wind farm. It is important to study these effects to find the optimal wind blade and wind farm configurations. The movie shows a blade turning from a wind source, and also shows the effect of the blade has on the wind field after the blade. The different views show the blade and wind field from the front, side, and top over time. Streamlines depict the change in the instantaneous wind field, which are colored by the wind vorticity. Blue is zero vorticity and red is greater than or equal to 1.0 vorticity.

SciDAC 2010

Web-Enabled Collaborative Climate Visualization 18

Wesley Kendall, Markus Glatter, Jian Huang, Dave Bernholdt, Forrest Hoffman



Snapshot of SeeESG, a collaborative interface for querying data from the Earth System Grid.

Video description:

The recent advances in high performance computing, storage and networking technologies have enabled fundamental changes in current climate research. While sharing datasets and results is already common practice in climate modeling, direct sharing of the analysis and visualization process is also becoming feasible. We show our efforts to develop a capability, coupled with the Earth System Grid (ESG), for sharing an entire executable workspace of visualization among collaborators. Evolutionary history of visualizations of research findings can also be captured and shared. The data intensive nature of the visualization system requires using several advanced techniques of visualization and parallel computing. With visualization clients implemented through standard web browsers, however, the ensuing complexity is made transparent to end-users. We demonstrate the efficacy of our system using cutting edge climate datasets.

SciDAC 2010

Comparing Manta Raycaster vs. OpenGL Rendering in ParaView with Metal Ball Splash Data

19

Li-Ta Lo, John Patchett, Jim Ahrens, Jon Woodring, Dave DeMarle, Nathan Fabian

Manta raycaster rendering in ParaView of a metal ball hitting a metal plate at high velocity.

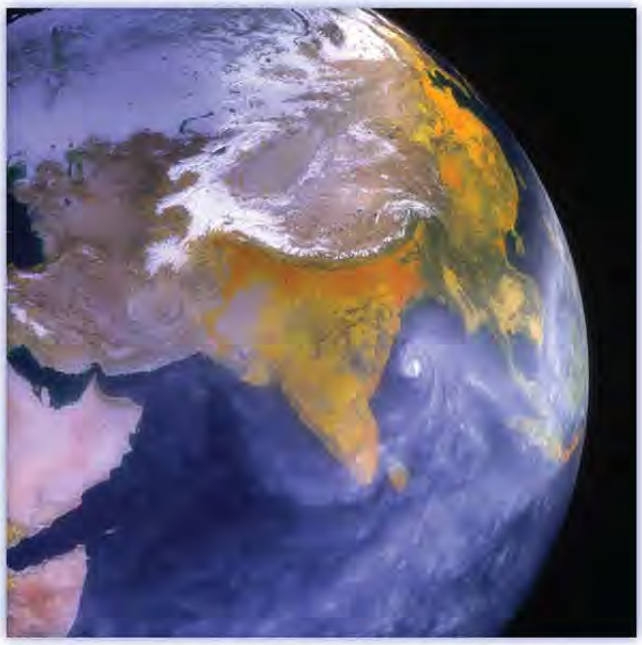
Video description:

At the petascale and going into exascale, visualization and rendering may be able to run efficiently on the supercomputing platform, rather than relying on visualization clusters with GPUs for rendering. To achieve high-performance rendering with supercomputing nodes, we have tested multi-core software ray-tracing for rendering, such as SCI's Manta raytracer. Initial results show that rendering software optimized for multi-core CPU provides competitive performance to GPU clusters for rendering of large data. Additionally, raytracing can provide additional advanced rendering techniques that basic OpenGL cannot easily provide, such as reflections and shadows. This movie provides a visual comparison between OpenGL rendering and Manta raytracing of a metal ball hitting a metal plate at high velocity, highlighting the advanced shading differences provided by raytracing. The Manta raytracing engine is integrated into VTK and ParaView as a render view, and it provides an alternative to OpenGL on multi-core systems.

SciDAC 2010

CCSM4 HOMME - TMQ (Water Vapor) / Landscan 20

Jamison Daniel, Mark Taylor, Kate Evans, Eddie Bright



The proper simulation of the distribution of water vapor in the climate system is essential to the accurate treatment of the hydrological cycle and the planetary radiation budget. These images show the simulated monthly-averaged distribution of the total column water vapor from a high-resolution configuration of the CCSM Community Atmospheric Model.

The LandScan Dataset comprises a worldwide population database compiled on a 30 arcsecond by 30 arcsecond latitude/longitude grid. LandScan has been developed as part of the Oak Ridge National Laboratory (ORNL) Global Population Project for estimating ambient populations at risk.

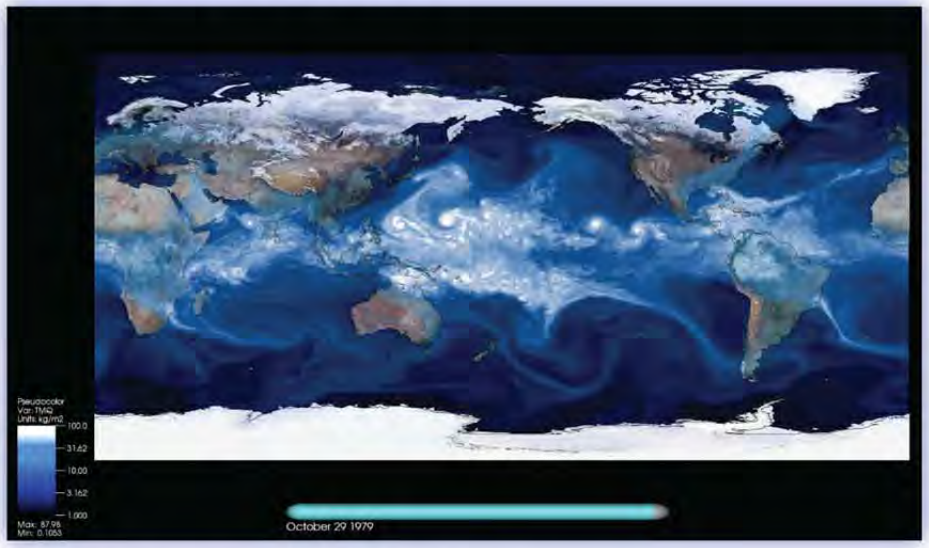
Video description:
HOMME: 1/8 degree resolution; Two-hourly; Two months (Dec. and Jan.). Original cubed-sphere gridded data was interpolated to a grid 25% higher resolution than the original run to avoid interpolation artifacts. CAM 3.5.1 'track 1' physics configuration, but with no surface roughness term in the gravity wave drag parametrization. The data is comprised of instantaneous fields. The LandScan data resolution is 30 arc seconds. The coordinates are Geographic WGS84. The LandScan dataset shown in the visualization is LandScan 2005.

SciDAC 2010

Hurricane Season

21

Prabhat, Michael Wehner, Wes Bethel



Tropical Cyclone formation and evolution.

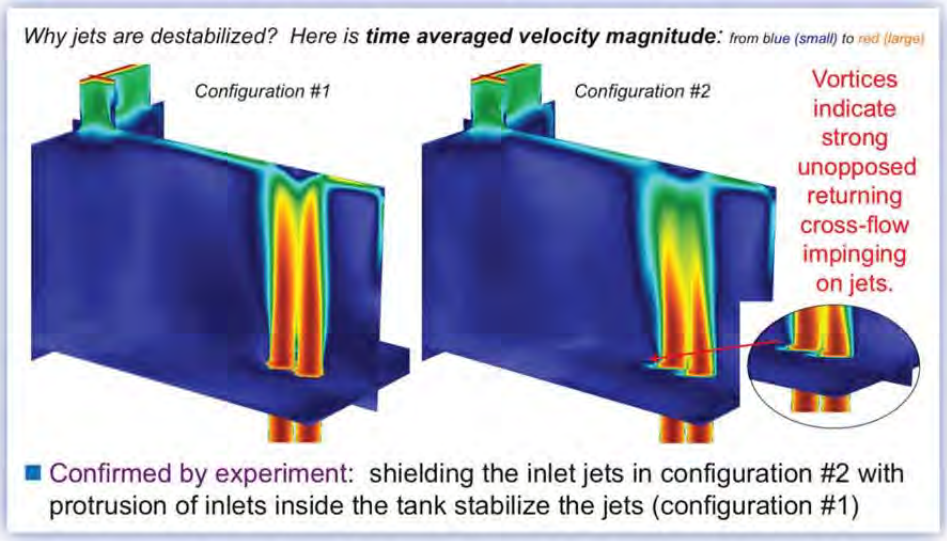
Video description:
Global warming will likely change the statistics of tropical cyclones and hurricanes. In this high resolution simulation, using the finite volume version of NCAR's Community Atmosphere Model, we are studying how well the model can reproduce observed tropical cyclone statistics. The simulated storms seen in this animation are generated spontaneously from the model's simulated weather conditions long after the memory of initial conditions have been forgotten. The structure of these simulated tropical cyclones is surprisingly realistic with the strongest storms rating as Category 4 on the Sapphir-Simpson scale. The animated field is the total vertical column integrated water vapor.

SciDAC 2010

Using Simulation and Visualization to Assess Experiment Configuration

22

A. Obabko, P. Fischer, A. Siegel, S. Lomperski, W. D. Pointer, E. Merzari



Time averaged velocity magnitude for two configurations of MAX experiment: small protrusion of inlets inside the tank (left) and inlets flushed to the bottom of the tank (right). The simulations and their visualization helped us to discover a jet destabilization mechanism.

Video description:

Using simulations and their visualization of two configurations of MAX experiment helped us to uncover a mechanism for jet destabilization resulted from minor changes in experiment geometry.

The MAX experiment is a mock-up of the upper plenum in sodium cooled fast reactors. It was build at ANL to study gas mixing and heat transfer, and to validate numerical codes for thermal striping problem.

We have conducted numerical simulations of turbulent flow in two different configurations of the experiment with Nek5000 code. The visualization of the simulations with VisIt helped us to discover the physical mechanism that led to destabilization of the inlet jets in the flow after minor changes to experiment configuration.

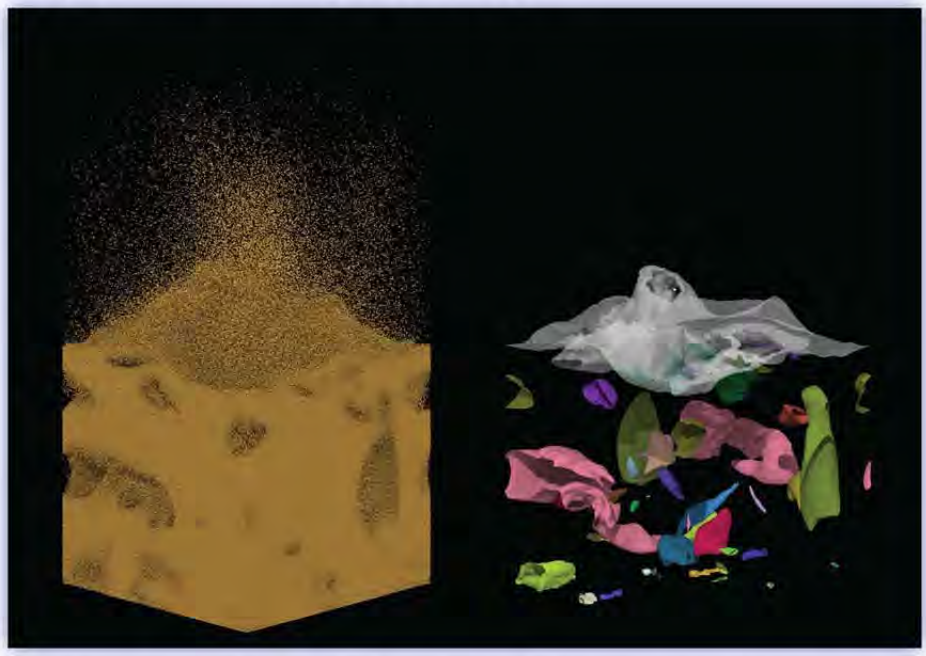
Our recommendations for stabilization of the flow were successfully implemented in the experiment.

SciDAC 2010

Particle Dynamics in a Fluidized Bed Reactor

23

Kenny Gruchalla, Perrine Pepiot, Olivier Desjardins



Left: A rendering of a periodic bed of sand fluidized by a gas stream injected from below. Right: Individual bubbles defined by a gas volume fraction of greater than 90% have been extracted and segmented from the particle bed.

Video description:

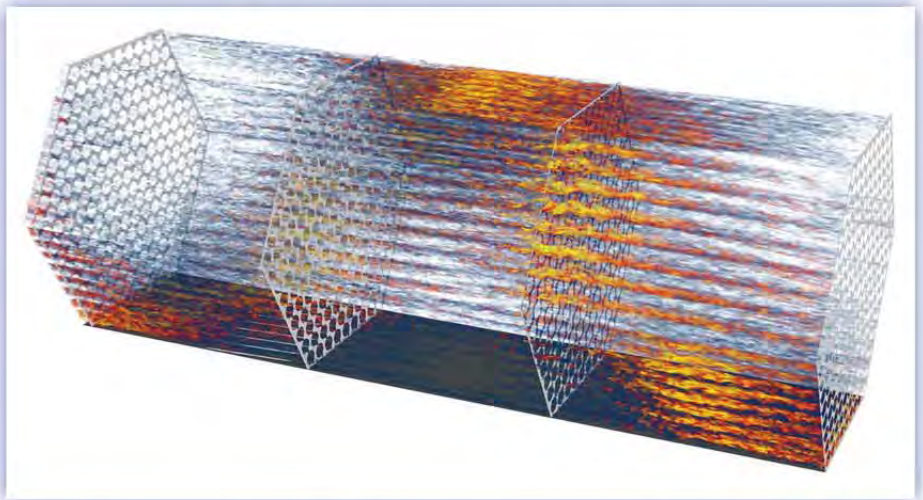
Fluidized bed reactors are a promising technology for the thermo-chemical conversion of biomass in biofuel production. However, the current understanding of the behavior of the materials in a fluidized bed is limited. We are using high-fidelity simulations to better understand the mechanics of the conversion processes. This video visualizes a simulation of a periodic bed of sand fluidized by a gas stream injected at the bottom. A Lagrangian approach describes the solid phase, in which particles with a soft-sphere collision model are individually tracked along the reactor. A large-scale point-particle direct numerical simulation involving 12 million particles and 4 million mesh points has been performed, requiring 512 cores for 4 days. The onset of fluidization is characterized by the formation of several large bubbles that rise and burst at the surface and is followed by a pseudo-steady turbulent motion showing intense bubble activity.

SciDAC 2010

Visualizing Coolant Flow in Sodium Fast Reactor Subassemblies

24

Hank Childs, Janet Jacobsen, Paul Fischer, Aleks Obabko



This volume rendering shows areas of high speed (colored yellow and red) through a simulation of a 217 pin reactor on a one billion grid point unstructured mesh.

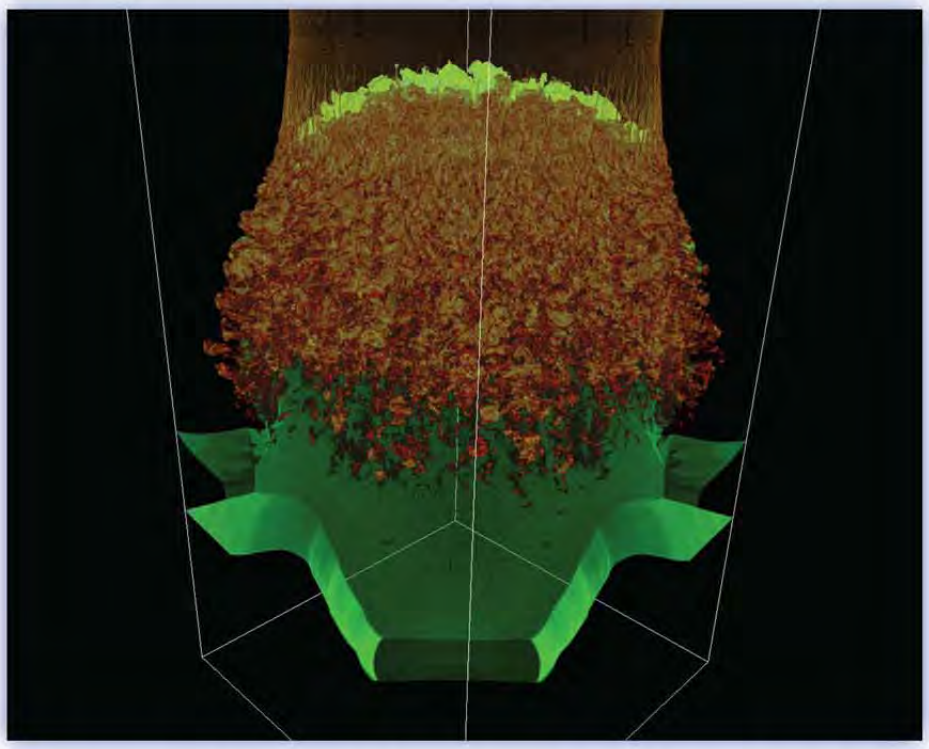
Video description:

Interchannel cross-flow is the principal cross-assembly energy transport mechanism for sodium fast reactors. For this design, uniform temperatures enable peak power output. This Nek5000 simulation models coolant flow for a 217-pin subassembly, using an unstructured mesh with over one billion grid points, resulting in five billion degrees of freedom per time slice. This visualization of the flow distribution, done with VisIt, provides the type of insight required for improving subassembly designs.

SciDAC 2010

Flash Code Simulation of a University of Michigan High Energy Density Blast-Wave Driven Rayleigh-Taylor Experiment 25

Nathan Hearn, Tomasz Plewa, Jonathan Gallagher



This image shows the simulated growth of instabilities in the dense-to-light interface of the University of Michigan NLUF OMEGA Laser experiment.

Video description:

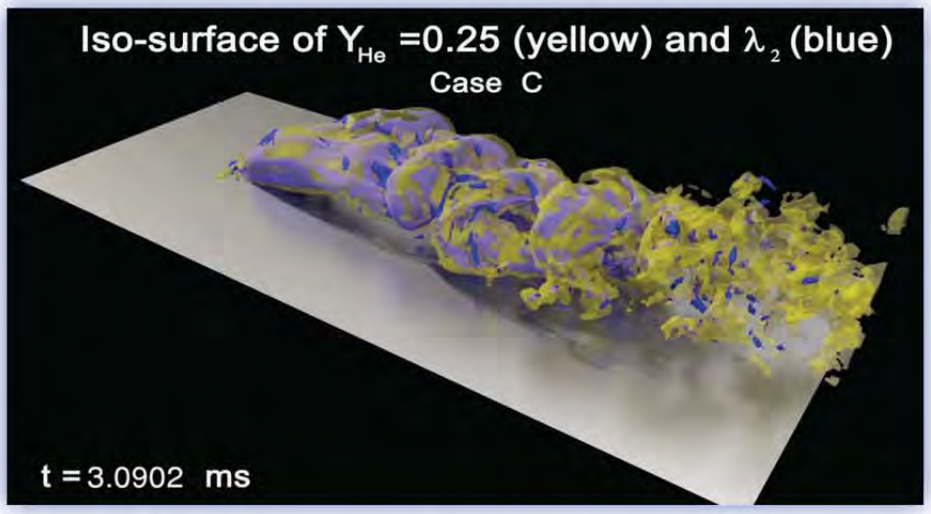
This movie shows the simulated growth of instabilities in the dense-to-light interface of the University of Michigan NLUF OMEGA Laser experiment. The cylinder is 900 microns in diameter and is resolved to 1.2 microns. The laser input is taken from a 2D, axisymmetric HYADES simulation and mapped onto the Flash code's AMR grid. The duration of the simulation is 20 ns in physical time.

SciDAC 2010

LES of an Inclined Jet into a Supersonic Turbulent Cross-Flow

26

Antonino Ferrante, Georgios Matheau, Paul Dimotakis, Richard Walters



An isosurface of Vorticity computed via Lambda2 combined with He Mass Fraction.

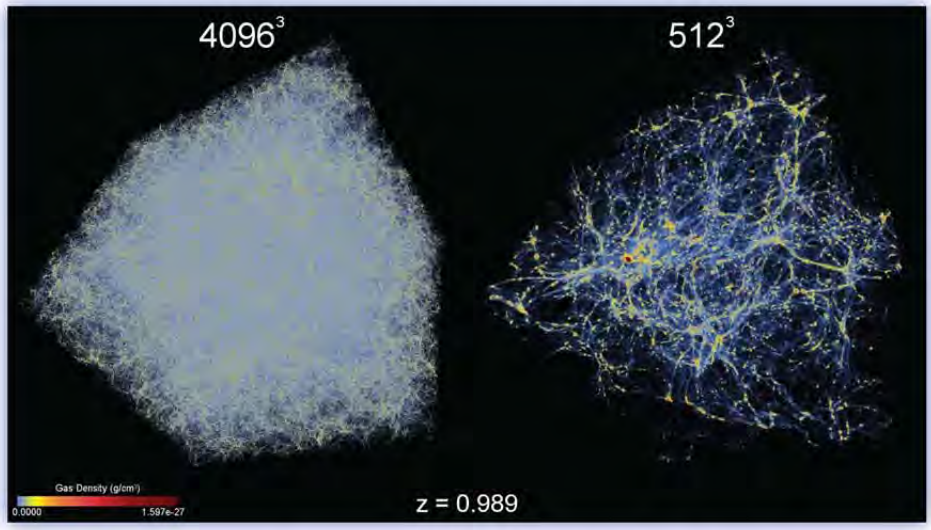
Video description:
A 3.23mm jet injects He into a high velocity flow.

SciDAC 2010

Searching for Baryon Acoustic Oscillations in Intergalactic Absorption, Project StarGate Application Driver 27

Science
Michael L. Norman, Robert Harkness, Pascal Paschos, Rick Wagner

Visualization
Mark Hereld, Joseph A. Insley, Eric C. Olson, Michael E. Papka



This figure shows side by side views of the full 4096^3 and a 512^3 sub-region of the density field from the baryon acoustic oscillations simulation.

Video description:

This simulation follows the growth of density perturbations in both gas and dark matter components in a volume 1 billion light years on a side beginning shortly after the Big Bang and evolved to half the present age of the universe. It calculates the gravitational clumping of intergalactic gas and dark matter modeled using a computational grid of 64 billion cells and 64 billion dark matter particles. The simulation uses a computational grid of 4096^3 cells and took over 4,000,000 CPU hours to complete.

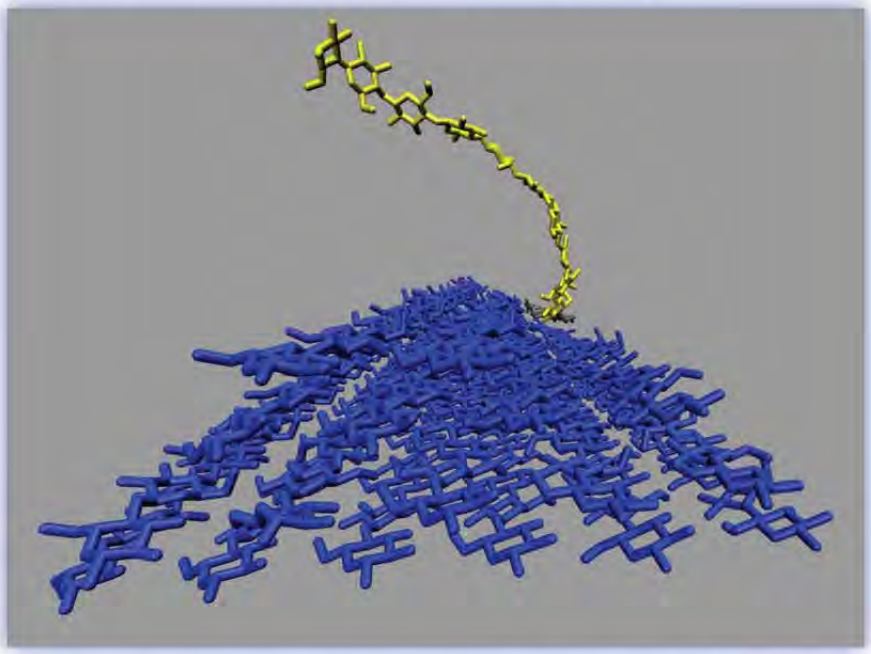
The visualization shows the density value, whose range spans over 6 orders of magnitude. The animation of the evolution was made from 148 terabytes of data generated on Kraken at the National Institute for Computational Sciences and transferred to Argonne over ESnet. The rendering was performed on Eureka at the Argonne Leadership Computing Facility.

SciDAC 2010

De-crystallization of Cellulose

28

Gregg T. Beckham, James F. Matthews, Michael F. Crowley



Decrystallization of cellulose simulation as a function of the reaction coordinate.

Video description:

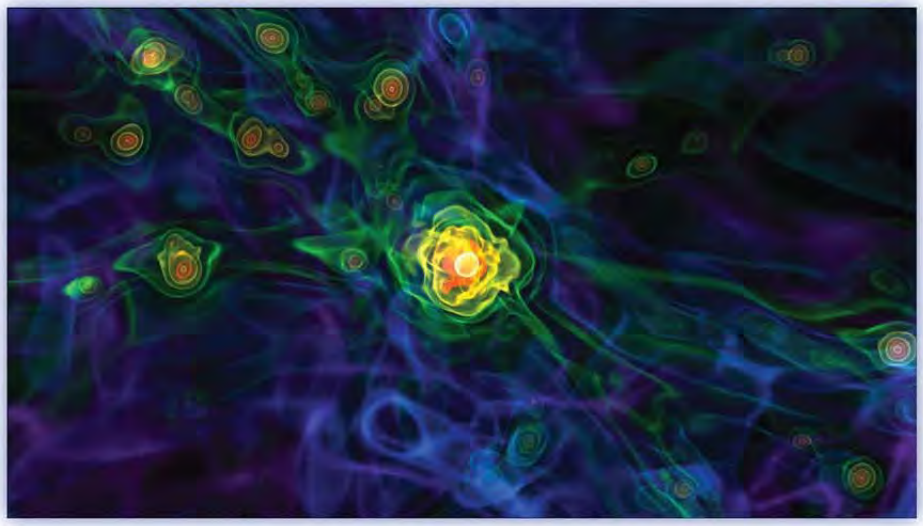
Cellulose-degrading organisms in the biosphere have evolved a complex cocktail of enzymes for overcoming biomass recalcitrance. Most enzyme cocktails contain both processive and non-processive cellulases, which are able to de-crystallize a single cellulose strand and hydrolyze the glycosidic linkages along a cellulose chain. A ubiquitous step that both processive and non-processive cellulases must accomplish is cellulose de-crystallization, for which there is an inherent thermodynamic penalty. Here, we measure the thermodynamic cost for cellulose de-crystallization as a function of cellulose morphology at the atomic scale with molecular simulation. Using molecular dynamics and umbrella sampling we measure the free energy cost of de-crystallizing each cellobiose unit. Our results indicate how enzymes may have evolved to degrade biomass in the biosphere at the molecular-level and suggest directions for increasing the enzymatic accessibility for cellulose digestion in biofuels applications.

SciDAC 2010

Evolution of a Galaxy Cluster in Adaptive Mesh Refinement Cosmological Simulations

29

Samuel W. Skillman, Brian W. O'Shea, Matthew J. Turk



Density isocontours of a galaxy cluster in a cosmological simulation.

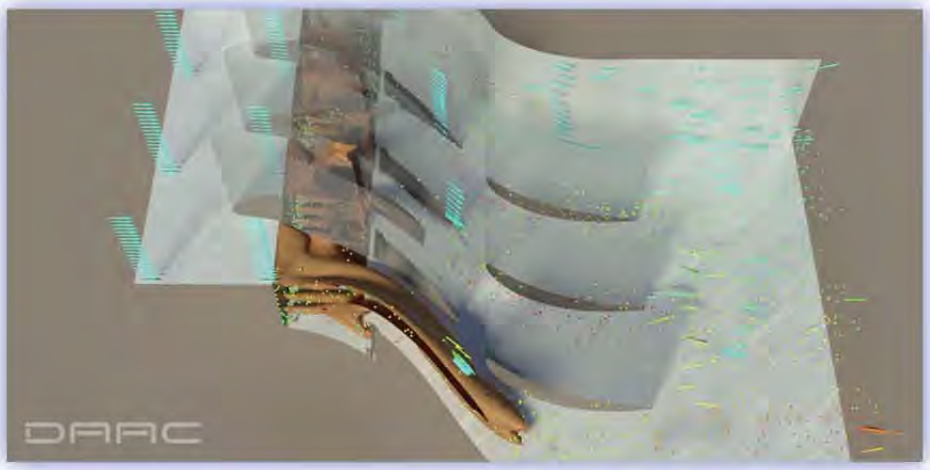
Video description:

This movie shows density isocontours of a galaxy cluster during its formation and evolution over 10 billion years. This shows the complex behavior associated with merging clusters, shocks, and virialization.

SciDAC 2010

Effect of Trapped Vortex Combustion with Radial Vane Cavity Arrangements on Predicted Inter-Turbine Burner Performance 30

Balu Sekar, Richard Walters, Kevin George, Miguel Valenciano



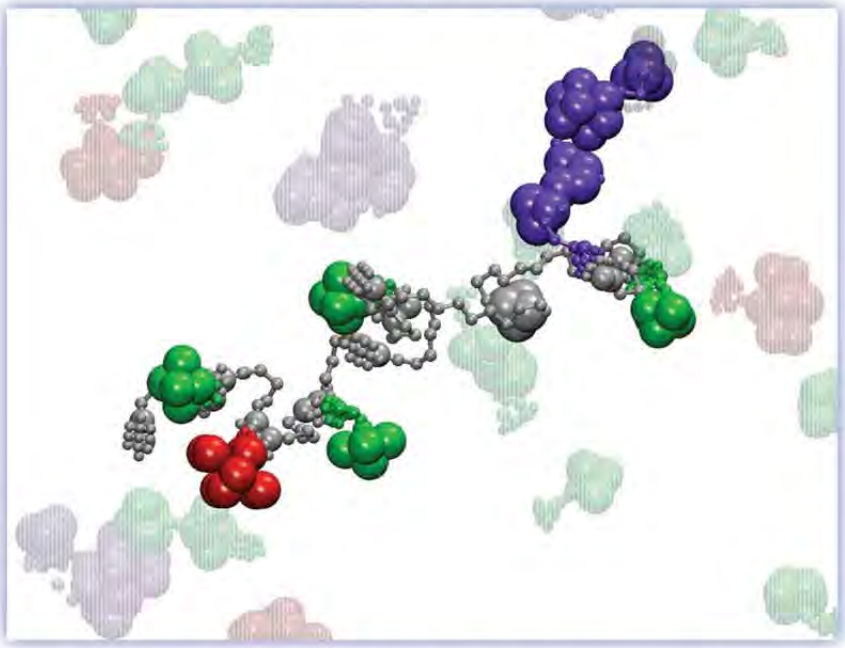
Visualization of flow through the new turbine design.

Video description:
Complex combustion processes, including chemical reactions, turbulence, unsteady, multiphase flow, evaporation and head and mass transfer pose great challenges in modern propulsion system design. AFRL has proposed placing an Ultra-Compact combustor (UCC) between a high pressure turbine stage and low pressure turbine stage to create an innovative Inter-Turbine Burner (ITB) concept. This work focuses on the evaluation of ITB combustor technologies that make use of HPC simulation.

SciDAC 2010

Cellulosomal Enzyme Complex Self-Assembly 31

Michael F. Crowley, Yannick J. Bomble, Michael E. Himmel



Cellulosomal enzymes assembling on a scaffolding protein. The red, green, and purple groups are different types of enzymatic subunits that bind to the scaffolding and break the bonds of sugar chains in plant cell walls.

Video discription:

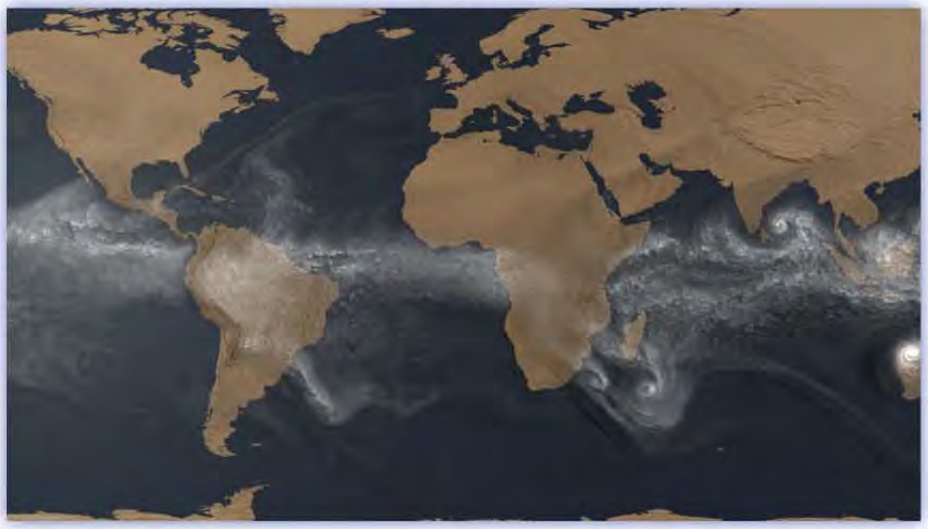
Cellulosomes are large multiprotein complexes that can exceed hundreds of millions of atoms in size, making them too massive for atomistic modeling with current computational technologies. Coarser, mesoscale models that describe interactions among structural subunits of the cellulosome rather than individual atoms are used to study cellulosomes. The image shows a mesoscale model of cellulosome assembly developed at NREL and BESC using software developed in SciDAC. The gray "scaffolding" protein forms the structural backbone of this simple cellulosome. The red, green, and purple groups are different types of enzymatic subunits that bind to the scaffolding and break the bonds of sugar chains in plant cell walls. This is a box of sixty enzymes, the ones that bind in the simulation are rendered as solid. [Courtesy of NREL researchers Michael F. Crowley, Yannick J. Bomble, and Michael E. Himmel, National Renewable Energy Laboratory, the BioEnergy Science Center, and SciDAC]

SciDAC 2010

HOMME — Eastern Hemisphere Typhoon Season — Dec/Jan

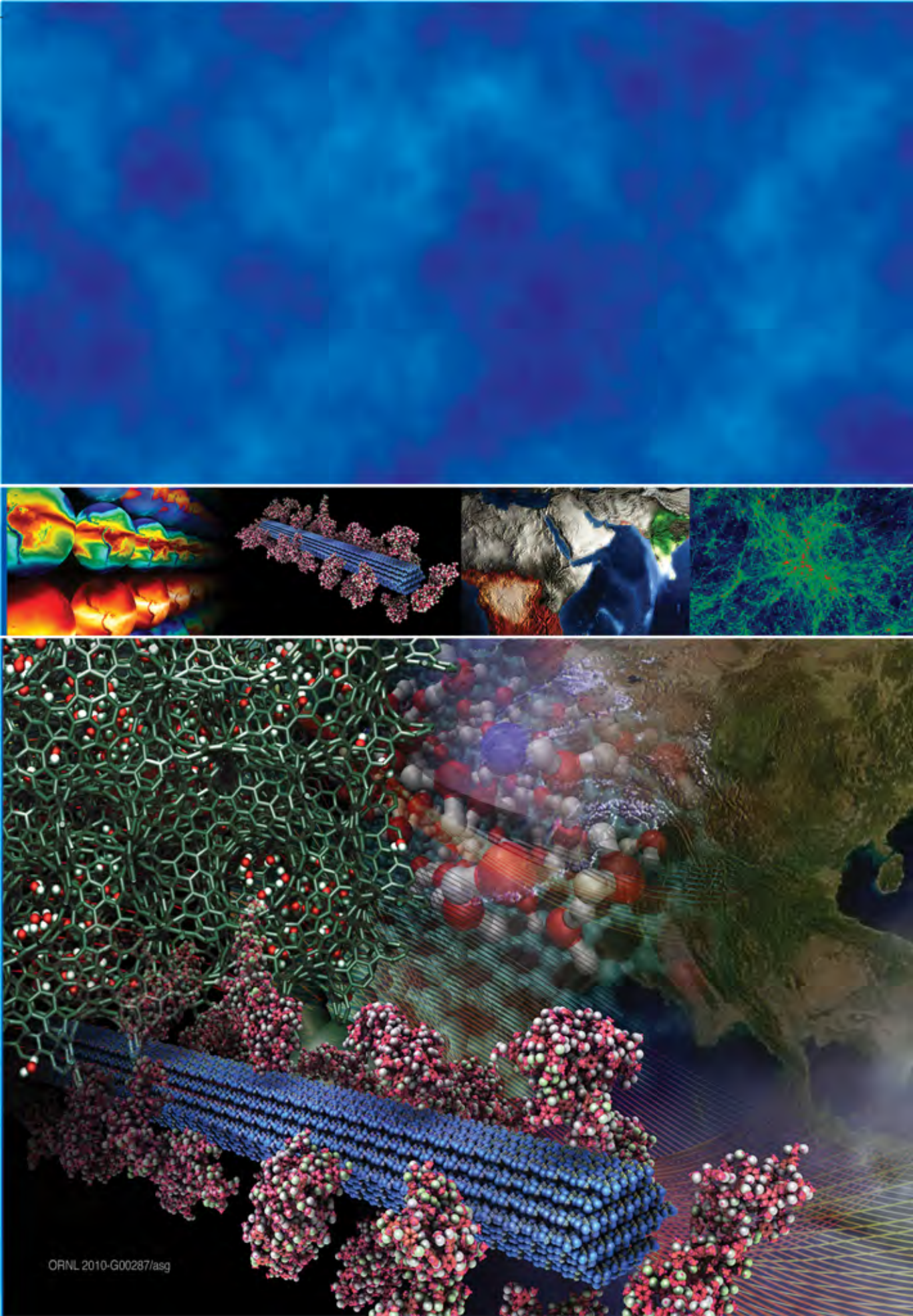
32

Jamison Daniel, Mark Taylor, Kate Evans, Jim Hack



Video description:
Animation of a present day 1/8 degree resolution of columnar precipitable water in the atmosphere. Note that regional features such as cyclones and diurnal precipitation cycles can be discerned at this resolution. This simulation is from CCSM4 with the new spectral element atmospheric dynamical core in a "time-slice" configuration: active atmosphere, land and ice components with prescribed ocean conditions and sea ice extent.

SciDAC 2010


ORNL 2010-G00287/asg