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.

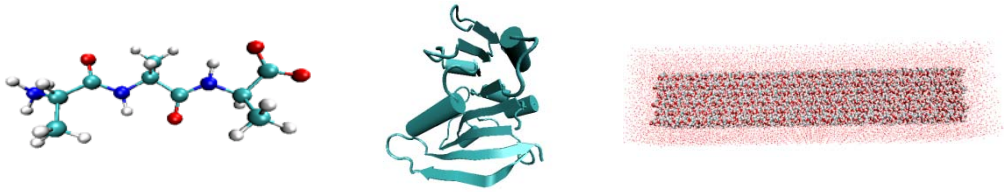

Figure 1. The test suite for validation included, from left to right, trialanine peptide with 33 atoms, DHFR enzyme with 2489 atoms, and a cellulose crystal fiber with 88,101 atoms solvated in TIP3P water.

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.