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 (UltavVis/Sandia), K Ma(UltavVis/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-β scheme used to discretize the time.

The electromagnetic fields generated by an electric current density \(\mathbf{J}\) 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:

\[
\nabla \times \left( \frac{1}{\mu} \nabla \times \mathbf{E}(\mathbf{r}, t) \right) + \varepsilon \frac{\partial^2 E(\mathbf{r}, t)}{\partial t^2} + \sigma \frac{\partial E(\mathbf{r}, t)}{\partial t} = -\frac{\partial}{\partial t} \mathbf{J}(\mathbf{r}, t), \text{ in } V \tag{1}
\]

where \(\varepsilon\), \(\mu\), and \(\sigma\) 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 \(S_R\):
where \( \mathbf{n} \) represents the outward unit vector normal to \( S \), \( Y \) denotes the surface admittance of the boundary \( S_R \), and \( \mathbf{U}(\mathbf{r}, t) \) 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 \( \mathbf{E}(\mathbf{r}, t) \), denoted by \( \mathbf{E}(\mathbf{r}, t) \equiv \int_{-\infty}^{t} \mathbf{E}(\mathbf{r}, \tau) \, d\tau \), and solve the following curl-curl equation instead [5]

\[
\mathbf{n} \times \left( \frac{1}{\mu} \mathbf{n} \times \mathbf{E}(\mathbf{r}, t) \right) + Y \mathbf{n} \times \frac{\partial}{\partial t} \mathbf{E}(\mathbf{r}, t) = \mathbf{U}(\mathbf{r}, t) \quad \text{on} \quad S_R
\]

We discretize the field using tangentially continuous \( \mathbf{N} \)édec basis functions [14], \( \mathbf{E}(\mathbf{r}, t) = \sum_{i=1}^{N} x_i(t) \mathbf{N}_i(\mathbf{r}) \), with \( N \) denoting the total number of unknowns, and obtain the equation in the matrix and vector form:

\[
\mathbf{K} \mathbf{x} + \mathbf{M} \frac{\partial^2}{\partial t^2} \mathbf{x} + (\mathbf{r} + \mathbf{Q}) \frac{\partial}{\partial t} \mathbf{x} = \mathbf{f}
\]

where \( \mathbf{x} = (x_1, x_2, ..., x_N)^T \) and \( \mathbf{K}, \mathbf{M}, \mathbf{r}, \mathbf{Q} \) are \( N \)-by-\( N \) square matrices given by

\[
\mathbf{K}_{ij} = \int_V \frac{1}{\mu} \left( \nabla \times \mathbf{N}_i(\mathbf{r}) \right) \cdot \left( \nabla \times \mathbf{N}_j(\mathbf{r}) \right) \, dV
\]

\[
\mathbf{M}_{ij} = \int_V \varepsilon \mathbf{N}_i(\mathbf{r}) \cdot \mathbf{N}_j(\mathbf{r}) \, dV
\]

\[
\mathbf{r}_{ij} = \int_V \sigma \mathbf{N}_i(\mathbf{r}) \cdot \mathbf{N}_j(\mathbf{r}) \, dV
\]

\[
\mathbf{Q}_{ij} = \int_{S_R} Y \mathbf{n} \cdot \mathbf{N}_i(\mathbf{r}) \cdot \left( \mathbf{n} \times \mathbf{N}_j(\mathbf{r}) \right) \, dS
\]

The vector \( \mathbf{f} \) is the driving force that is given by

\[
f_i = - \int_V \mathbf{N}_i(\mathbf{r}) \cdot \mathbf{f}(\mathbf{r}, t) \, dV - \int_{S_R} \mathbf{N}_i(\mathbf{r}) \cdot \left( \int_{-\infty}^{t} \mathbf{U}(\mathbf{r}, \tau) \, d\tau \right) \, dS
\]

In solving the above equation, the implicit Newmark-\( \beta \) scheme [15] is employed for numerical time integration. The method is proven to be unconditionally stable when the prescribed parameter \( \beta \) is larger than or equals to 0.25. With the Newmark-\( \beta \) time discretization, a system of linear equations is needed to be solved at each time step:

\[
[M + \frac{\varepsilon \Delta t}{2} (\mathbf{r} + \mathbf{Q} + \beta (c \Delta t)^2 \mathbf{K})] \mathbf{x}^{n+1} = \mathbf{b}
\]

with \( \mathbf{b} = \frac{[2M - (1 - 2\beta)(c \Delta t)^2 \mathbf{K}] \mathbf{x}^n}{c \Delta t^2} - \left[ M - \frac{1}{2}c \Delta t(\mathbf{r} + \mathbf{Q} + \beta (c \Delta t)^2 \mathbf{K}) \right] \mathbf{x}^{n-1} \)
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 $\vec{E}$ and the magnetic flux density $\vec{B}$ can be computed from the solution vector $\mathbf{x}$:

$$\vec{E}(\vec{r}) = \sum_i \frac{\partial}{\partial t} x_i \vec{N}_i(\vec{r}) \quad (13)$$

$$\vec{B}(\vec{r}) = -\sum_i x_i \nabla \times \vec{N}_i(\vec{r}) \quad (14)$$

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:

$$W(\vec{p}', \vec{p}, s) = -\frac{1}{q} \int_{z_1}^{z_2} E_q(\vec{p}, z, t)|_{k=x \pm i} \, dz \quad (15)$$

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 $|\vec{p}'|$ and $\vec{p}$, 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 $s_{\text{max}}$ of the wakefield that needs to be computed. Namely, $b + d >= s_{\text{max}}$, 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.

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](image)

**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 $x^n$, $x^{n-1}$, $f^n$, and $f^{n-1}$ 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 $e^{old}$ with a set of basis functions $\{N_1^{old}\}$ on an unstructured mesh, we need to obtain the discretized field $e^{new}$ with a different set of basis functions $\{N_i\}$ on a new mesh that accurately represents the same field $e^{old}$ on the old mesh. Algorithm 1 described the details of the method. Note that $\alpha$ 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 $\alpha$ 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 recommended to use $\alpha = \frac{1}{4} (c\Delta t)^2$ in the simulation where $c$ is the speed of light and $\Delta t$ 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 a 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
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. 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 $\sigma = 20$ 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 $\sigma = 0.6$ 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 $\sigma = 20 \text{ mm}$ 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 $b = 0.1 \text{ m}$ and $f = 0.2 \text{ m}$. The initial mesh size in the first run is 10 mm. The second run uses traditional FETD approach with a mesh size of 5 mm.

Figure 6. Wake potential of a $\sigma = 0.6 \text{ mm}$ 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.
**Figure 8.** Wake potential of a $\sigma = 0.3$ 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**


