SLAC-PUB-14099

A moving window technique in parallel finite element time domain electromagnetic simulation

Lie-Quan Lee*, Arno Candel, Cho Ng, Kwok Ko

SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025

Abstract

A moving window technique for the finite element time domain (FETD) method is developed to simulate the propagation of electromagnetic waves induced by the transit of a charged particle beam inside large and long structures. 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 using the moving window 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.

Keywords:
Moving window, mesh refinement, FETD, unstructured mesh, wakefield

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

*Corresponding author.
Email address: liequan@slac.stanford.edu (Lie-Quan Lee)

Work supported in part by US Department of Energy contract DE-AC02-76SF00515.

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 computational tool to calculate wakefields for large accelerator structures using the finite element time domain method [4, 5]. 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 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 $\vec{\mathbf{J}}$ 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:

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

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:

$$\begin{aligned} \vec{\mathbf{n}} \times \vec{\mathbf{E}}(\vec{\mathbf{r}},t) &= 0 \qquad on \quad S_E \quad (2)\\ \vec{\mathbf{n}} \times \left(\frac{1}{\mu}\nabla \times \vec{\mathbf{E}}(\vec{\mathbf{r}},t)\right) &= 0 \qquad on \quad S_M \quad (3)\\ \vec{\mathbf{n}} \times \left(\frac{1}{\mu}\nabla \times \vec{\mathbf{E}}(\vec{\mathbf{r}},t)\right) + Y\vec{\mathbf{n}} \times \vec{\mathbf{n}} \times \frac{\partial}{\partial t}\vec{\mathbf{E}}(\vec{\mathbf{r}},t) &= \vec{\mathbf{U}}(\vec{\mathbf{r}},t) \quad on \quad S_R \quad (4) \end{aligned}$$

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 $\vec{\mathscr{E}}(\vec{\mathbf{r}}, t) \equiv \int_{-\infty}^{t} \vec{\mathbf{E}}(\vec{\mathbf{r}}, \tau)\, \mathrm{d}\tau$, and solve the following curl-curl equation instead [5]

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \vec{\mathscr{E}} \right) + \varepsilon \frac{\partial^2}{\partial t^2} \vec{\mathscr{E}} + \sigma \frac{\partial}{\partial t} \vec{\mathscr{E}} = -\vec{\mathbf{J}}, \tag{5}$$

We discretize the field using tangentially continuous Nédélec basis functions [11], $\vec{\mathscr{E}}(\vec{\mathbf{r}}, t) = \sum_{i=1}^{N} x_i(t) \vec{\mathbf{N}}_i(\vec{\mathbf{r}})$, with N denoting the total number of unknowns, and obtain the equation in the matrix and vector form:

$$\mathbf{K}\mathbf{x} + \mathbf{M} \frac{\partial^2}{\partial t^2} \mathbf{x} + (\mathbf{R} + \mathbf{Q}) \frac{\partial}{\partial t} \mathbf{x} = \mathbf{f} \tag{6}$$

where $\mathbf{x} \equiv (u_1, u_2, ..., u_N)^T$ and $\mathbf{K}, \mathbf{M}, \mathbf{R}, \mathbf{Q}$ are N-by-N square matrices given by

$$\mathbf{K}_{ij} = \int_V \frac{1}{\mu} \left(\nabla \times \vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \right) \cdot \left(\nabla \times \vec{\mathbf{N}}_j(\vec{\mathbf{r}}) \right) \mathrm{d}V \tag{7}$$

$$\mathbf{M}_{ij} = \int_V \varepsilon \vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \cdot \vec{\mathbf{N}}_j(\vec{\mathbf{r}})\, \mathrm{d}V \tag{8}$$

$$\mathbf{R}_{ij} = \int_V \sigma \vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \cdot \vec{\mathbf{N}}_j(\vec{\mathbf{r}})\, \mathrm{d}V \tag{9}$$

$$\mathbf{Q}_{ij} = \int_{S_R} Y \left(\vec{\mathbf{n}} \times \vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \right) \cdot \left(\vec{\mathbf{n}} \times \vec{\mathbf{N}}_j(\vec{\mathbf{r}}) \right) \mathrm{d}S \tag{10}$$

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

$$\mathbf{f}_i = -\int_V \sigma \vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \cdot \vec{\mathbf{J}}(\vec{\mathbf{r}}, t)\, \mathrm{d}V - \int_S \vec{\mathbf{N}}_i(\vec{\mathbf{r}}) \cdot (\int_{-\infty}^{t} \vec{\mathbf{U}}(\vec{\mathbf{r}}, \tau)\, \mathrm{d}\tau)\, \mathrm{d}S \tag{11}$$

In solving the above equation, the implicit Newmark-β scheme [12] is employed for numerical time integration. The method is proven to be unconditionally stable when the prescribed parameter β is larger than or equals to 0.25. With the Newmark-β time discretization, a system of linear equations is needed to be solved at each time step:

$$\begin{aligned} & \left[\mathbf{M} + \tfrac{c\Delta t}{2}(\mathbf{R} + \mathbf{Q}) + \beta (c\Delta t)^2 \mathbf{K}\right] \mathbf{x}^{n+1} = \mathbf{b} \qquad (12) \\ \textit{with} \quad \mathbf{b} \equiv \quad & \left[2\mathbf{M} - (1 - 2\beta)(c\Delta t)^2 \mathbf{K}\right] \mathbf{x}^{n} \\ & - \left[\mathbf{M} - \tfrac{1}{2} c\Delta t (\mathbf{R} + \mathbf{Q}) + \beta (c\Delta t)^2 \mathbf{K}\right] \mathbf{x}^{n-1} \\ & -(c\Delta t)^2 \left[\beta \mathbf{f}^{n+1} + (1 - 2\beta)\mathbf{f}^{n} + \beta \mathbf{f}^{n-1})\right] \end{aligned}$$

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 $\mathbf{x}$:

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

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

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 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.

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. 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 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.

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 2. 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 2. The

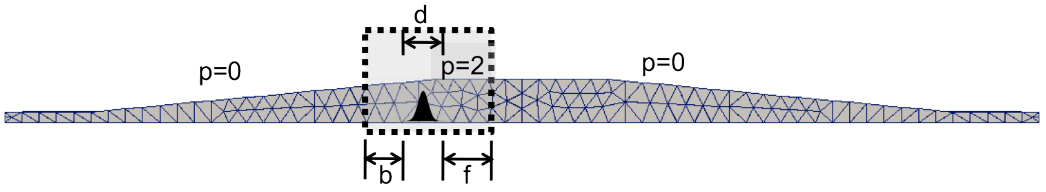


Figure 1: Moving window on an unstructured grid. The dashed box illustrates the moving window region. Its length is defined by $b + d + f$, where b is the length of the padded zone behind the beam, d the beam size and f the length of the padded zone in front of the beam. b and f are adjustable parameters in each specific problem. The particle beam moves from left to right.

new mesh is partitioned using ParMetis [13] and the elements in the mesh are redistrsibuted to corresponding processes. Then, the matrices in Equations (7 - 10) are re-assembled.

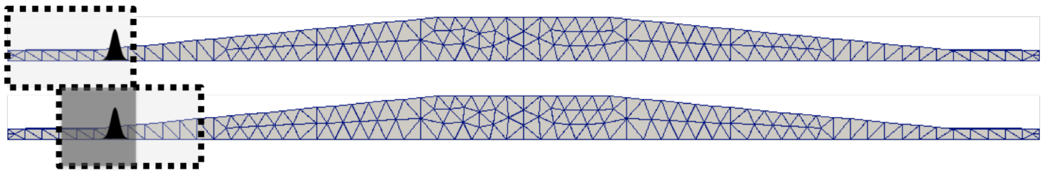


Figure 2: The window is moved along the beam transit direction when the beam front reaches the right boundary of the window. The dashed boxes illustrate the moving window regions. The top and bottom plots represent two consecutive windows. The shaded region is the overlapping computational domain between 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 $\mathbf{f}^{n-1}$ need to be transferred from the old mesh onto the new mesh. Algorithm 1 describes the details of transferring a vector on the old mesh and forming a new vector on the new mesh in a scalable way. The algorithm is based on the fact that the mesh elements in the overlapping computational domain do not change although they may be redistributed to different processes in parallel simulation. The vectors $\mathbf{x}^n$ and $\mathbf{x}^{n-1}$ are transferred using Algorithm 1. The vector $\mathbf{f}^n$ and $\mathbf{f}^{n-1}$ can be either transferred in a similar way or recomputed with the driving source at the corresponding time steps.

Algorithm 1 Transfer a vector from the old mesh to the new mesh.

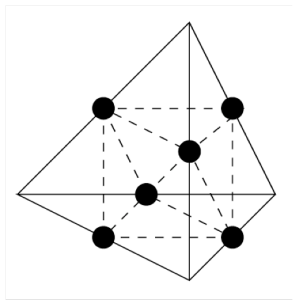
```
input: given a vector $\mathbf{v}$ on the old mesh
output: a new vector $\mathbf{v}'$ on the new mesh
condition: existence of an overlapping mesh region

  1. scatter values in the vector $\mathbf{v}$ to each mesh elements according to its
     associated degrees-of-freedom (DOF) information
       (a) keep the values when the mesh element is in the overlapping region
       (b) drop the element and its values when the element is outside the
           new window
       (c) set the values of the newly-added mesh elements to zeroes
  2. redistribute mesh elements in the new mesh together with their as-
     sociated values to different processes according to partitioning of the
     mesh
  3. gather values in each mesh element to form a new vector $\mathbf{v}'$ on the new
     mesh according to the updated DOF information
```

2.3. Moving Window with Mesh Refinement

The element size used in a simulation has to be small enough to resolve the frequency content of a particle beam; otherwise the results will not be accurate. For an ultra-short particle beam, the small element size required will lead to a prohibitively large number of mesh elements for the whole geometry to be generated and stored in the computer memory. Meanwhile, wakefield computation only requires a certain part of the mesh at a given time step, and it is this part that needs calculation accuracy. Thus, it is advantageous to start with a coarse mesh and refine the mesh inside the window as needed. In this section, we discuss how online mesh refinement is done to provide enough resolution for an ultra-short particle beam passing through a structure.

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 [14]. 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 3 illustrates the edge splitting in subdividing a linear or curvilinear tetrahedron.


(a) Subdividing a linear tetrahedron.

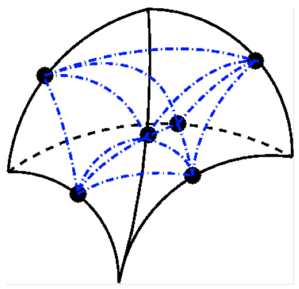

(b) Subdividing a curvilinear tetrahedron.

Figure 3: Subdividing tetrahedral elements by splitting all the six edges.

Once the refined mesh is created, the processes of mesh partitioning, mesh distribution, and matrix assembly can be easily done on the newly-created mesh. The difficult task is how to transfer the vectors $\mathbf{x}^n$, $\mathbf{x}^{n-1}$, $\mathbf{f}^n$ and $\mathbf{f}^{n-1}$ that are only meaningful on the old mesh. Algorithm 1 cannot be applied any more because a newly-created tetrahedral element in the overlapping computational domain may have different vertex ordering. Thus, the tetrahedron may have an equivalent but different set of basis functions associated with it. A novel method [15] has been recently developed to project discretized electromagnetic fields from one mesh onto another. The details can be found in Reference [15]. 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 2 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 recommended to use $\alpha = \frac{1}{4}(c\Delta t)^2$ in the simulation where c is the speed of light and Δt the time step.

Algorithm 2 Transfer a vector from the old mesh to the new mesh with mesh refinement.

```
input: given a vector $\mathbf{e}^{old}$ on the old mesh
output: a new vector $\mathbf{e}^{new}$ on the new mesh
    1. define a field $\vec{\mathbf{F}}(\mathbf{e}^{old}) \equiv \sum_i e_i^{old} \vec{\mathbf{N}}_i^{old}$
    2. define the curl of the field $\nabla \times \vec{\mathbf{F}}(\mathbf{e}^{old}) = \sum_i e_i^{old} \nabla \times \vec{\mathbf{N}}_i^{old}$
    3. 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 $\alpha$ is an adjustable parameter
    4. solve the system of linear equations to obtain $\mathbf{e}^{new}$
       $(\mathbf{M} + \alpha\mathbf{K})\mathbf{e}^{new} = \mathbf{b}$
```

3. Examples and Results

In this section, we present two examples to demonstrate the effectiveness of using the moving window technique in the FETD method for wakefield calculations in accelertor applications. One uses the moving window without mesh refinement and the other with mesh refinement.

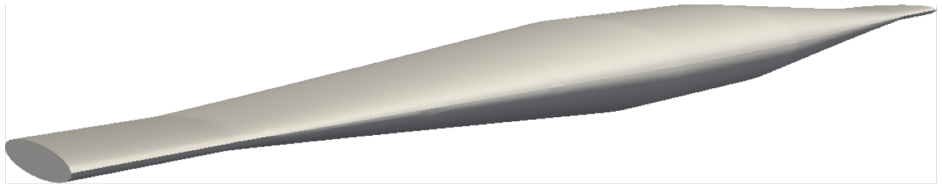


Figure 4: A computer model of an undulator taper structure. It provides smooth transitions from a 50 mm×6 mm elliptical pipe to a 75 mm×25 mm pipe then back to a 50 mm×6 mm pipe. The tapered transitions are 100 mm long on both sides. The center pipe is 50 mm long while the end pipes are 25 mm long. The total length of the model is 300 mm.

3.1. Example with p-refinement

In order to study wakefield effects on beam stability for storage ring accelerators, it is important to calculate the Green's function wakefield determined by a short particle bunch. For the storage ring PEP-X [1], currently under study as a new generation synchrotron light source, it is desirable to use a short bunch for the computation of wakefields of its beamline components.

In particular, the calculations for large and long beamline components will be computationally challenging at this bunch length. Such an example is the undulator structure with long smooth tapers connecting vacuum chambers with different cross sections. Figure 4 shows the computer model of the undulator structure. For all the wakefield calculations in thie paper, the excitation source is defined by a moving Gaussian bunch with RMS length σ, and in a typical run, a length of 10σ is used to cover the entire spatial extent of the bunch.

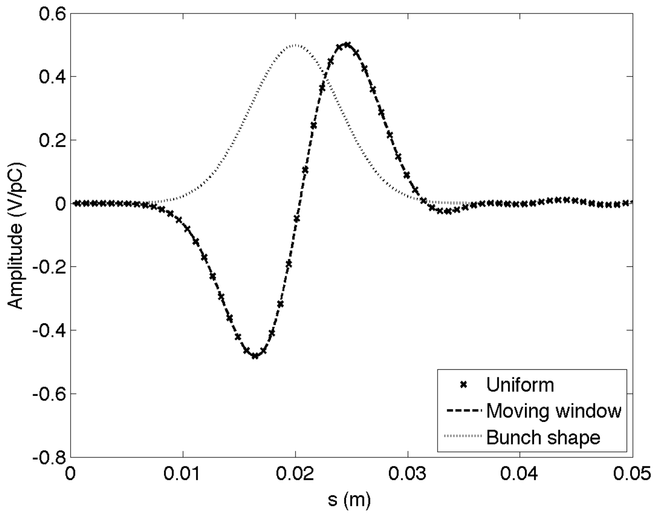


Figure 5: Wake potential of a $\sigma = 4$ mm bunch in the undulator structure. The dot-dashed curve represents the wakefield using the whole structure with 2 mm mesh size. The dashed curve represents the wakefield using a moving window. The results match for the first 50 mm as the beam size is 40 mm and the length of the trailing padded zone is 10 mm.

To verify the correctness of the moving window technique, we compare the simulations with and without moving window using a long bunch length, where the results can be computed readily without using substantial computer resources. We generated a tetrahedral mesh for the undulator taper model with 2 mm element size which is good enough to model the wakefield for a long $\sigma = 4$ mm bunch. Two simulations were carried out: the first run used the whole structure with 2nd order basis functions, and the second a moving window with 2nd order basis functions within it and zeroth order outside. In using the moving window, the length of the trailing padded zone

b is 10 mm and that of the front padded zone f 40 mm. Thus, the total window size is 90 mm since the beam size is 40 mm (10σ for a Gaussian beam with $\sigma = 4$ mm). The wakefields in Figure 5 show that the results from the two runs are identical, verifying the implementation of the moving window technique in the FETD simulation. Both runs were executed on 5 nodes of a Cray XT5 computer with each node containing two hex-core AMD Opteron processors, 16GB memory, and a SeaStar 2+ router. It took 21.5 minutes for the first run and only 11.0 minutes for the second run with a moving window. The moving window technique reduces the computational domain, the matrix and vector sizes associated with it, and thus the overall computational costs.

In calculating the wakefield for $\sigma = 0.5$ mm bunch, we generated a mesh with about 19 million quadratic tetrahedral elemesnts using 0.25 mm element size, which should be small enough to resolve the high frequency components of the short bunch. We set b to 60 mm, and f 25 mm so that the total window size is 90 mm. With this window size, the window moves 15 times until it is completely outside the undulator structure at the downstream. The maximum number of DOFs in all these consecutive windows is 54.9 millions. The simulation was performed on Jaguarpf [16] using 6000 cores and took 15 wall-clock hours. The wake potential is shown in Figure 6. It should be pointed out the wakefield for a Gausssian bunch can be constructed from that for a shorter Gaussian bunch. We used the wakefield shown in Figure 6 to constrcut excellently the wakefield in Figure 5 for a longer bunch, thus validating the calculation for the short bunch.

3.2. Example with h- and p-refinement

The second example is to calculate the short-range wakefield of a wiggler taper structure in PEP-X as shown in Figure 7. It provides smooth transitions from a 45 mm$\times$15 mm rectangular pipe to 48 mm circular beam pipe and back to a 45 mm$\times$15 mm rectangular pipe. The length of each transition is 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 laste 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

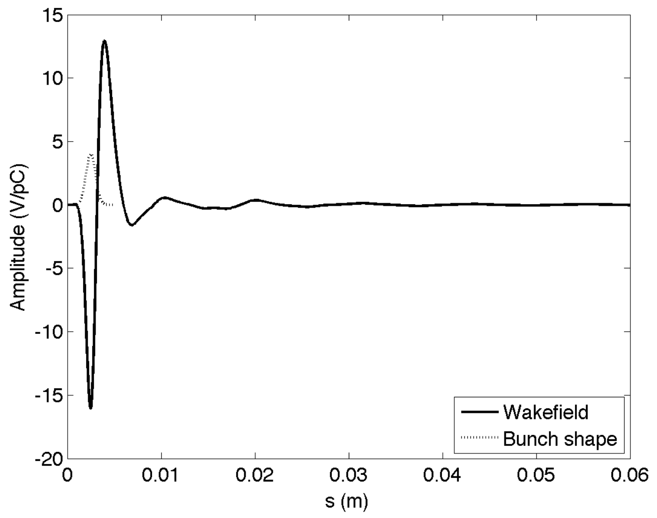


Figure 6: Wake potential of a $\sigma = 0.5$ mm bunch passing through an undulator taper, calculated with a 90 mm moving window.

size. For this coarser mesh, we ran the simulation using a moving window with one level of mesh refinement for the same beam size. In both runs, 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. It took about 23 minutes to finish the first run and about 36.5 minutes the second. The longer execution time in the second run is due to the overhead associated with mesh refinement. The wakefields from both runs are shown in Figure 8, and the results are in remarkable agreement for the first 0.3 m. This 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. Finally, the wake potential for a short particle beam with $\sigma = 1$ mm is plotted in Figure 9. It is calculated using a window with $b = 60$ mm and $f = 100$ mm. The initial coarse mesh was constructed using 1 mm element size. The meshes in consecutive windows were refined to 0.5 mm size elements during the computation. The window moves 11 times before it exits the downstream end of the structure. The maximum number of DOFs in all these windows is 74.7 millions. It took about 26 hours with 6000 cores of

Jaguarpf, a Cray XT5 to complete the simulation.

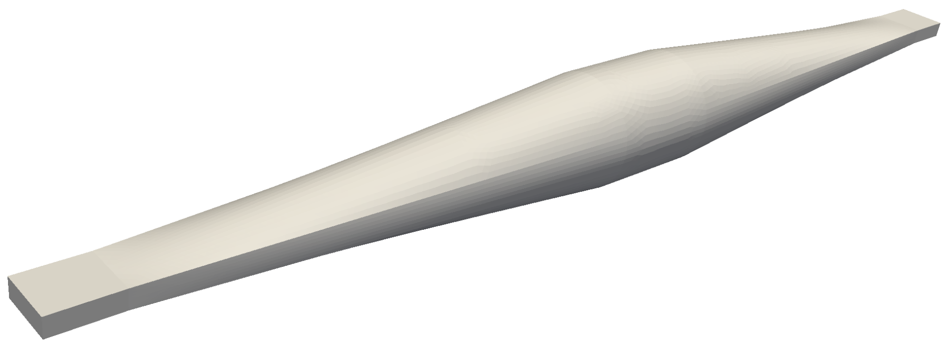


Figure 7: The side view of a computer model of the wiggler taper in PEP-X. It provides smooth transitions from a 45 mm$\times$15 mm rectangular pipe to 48 mm circular beam pipe and back to a 45 mm$\times$15 mm rectangular pipe. The transition length is 400 mm each. The circular beam pipe is 100 mm long and the rectangular beam pipes are both 50 mm long. The total length of the model is 1 m.

4. Concluding Remarks

A moving window finite element time domain (FETD) method is developed to expedite the simulate the propagation of electromagnetic waves induced by a particle beam transit through an electrically large and long structure. The efficiency of the moving window technique is achieved by using higher-order basis functions for the mesh elements inside the window (p-refinement). Additional gain in efficiency can be obtained by employing online mesh refinement in the moving window (h-refinement), especially when the simulation starts with a coarse mesh for the entire structure in reducing memory storage. Parallel algorithms for transferring discretized vectors from one mesh to another, which are the key to the success of the moving window technique, are presented. Numerical experiments have been carried out to validate the technique and demonstrate its advantanges over normal FETD computations in saving computing resources. It is shown that the moving window FETD method is an effective way to compute short-range wakefields of an ultra-short particle beam in large, complex accelerator structures.

Acknowledgements

The authors would like to thank Andreas Kabel for insightful discussions and Liling Xiao for providing the geometries of undulator and wiggler tapers.

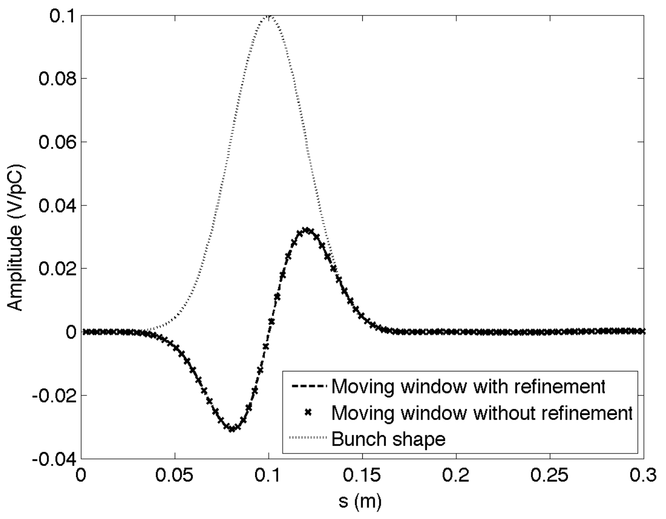


Figure 8: Comparison of wake potential due to a particle beam with $\sigma = 20$ mcalculated from two runs. The first run uses a moving window with mesh refinemet and the second without. The window size is 0.5 m in both cases with $b = 0.1$ m and $f = 0.2$ m. The initial mesh size in the first run is 10 mm and the second 5 mm.

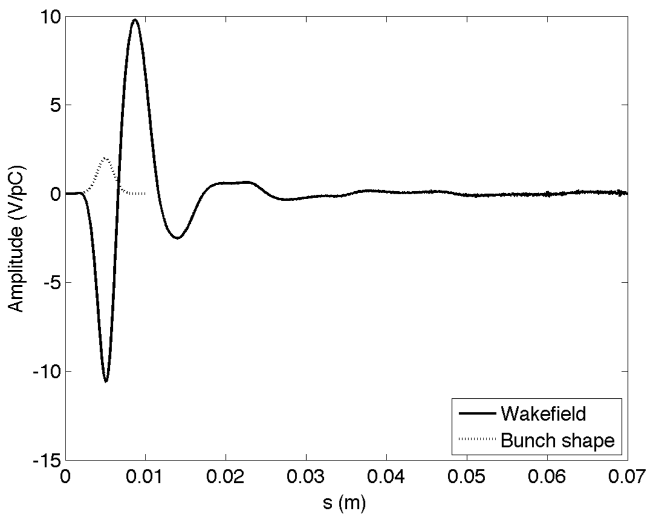


Figure 9: Wake potential of a σ =1 mm bunch passing through a wiggler taper structure, calculated with a 17 mm moving window and mesh refinement inside it.

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 . A design report of the baseline for PEP-X: an ultra-low emittance storage ring. Tech. Rep.; SLAC National Accelerator Laboratory; 2010.

[2] Zennaro R. Considerations on short range wakefield for CLIC main beam accelerating structures. Tech. Rep.; EUROTeV; 2007. EUROTeV-Report-2007-072.

[3] Hoffstaetter GH. Toward the energy recovery linac, a brighter x-ray source. 2005. URL `http://erl.chess.cornell.edu/`.

[4] Ko K, et al . SciDAC and the international linear collider. 2006. Proc. of the 2006 SciDAC Conference.

[5] Candel A, at el . Parallel higher-order finite element method for accurate field computations in wakefield and pic simulations. 2006. Proc. of the 9th International Computational Accelerator Physics Conference; URL `http://icap2006.web.cern.ch/icap2006/`.

[6] Candel A, Kabel A, Lee LQ, Li Z, Ng C, Schussman G, et al. State of the art in electromagnetic modeling for the compact linear collider. Journal of Physics Conference Series 2009;180(1):012004 –1.

[7] Ainsworth M. Dispersive properties of high-order Nedelec/edge element approximation of the timeharmonic Maxwell equations. Math phys eng sci 2004;362(1816):471–92.

[8] Fidel B, Heyman E, Kastner R, Ziolkowski RW. Hybrid ray-FDTD moving window approach to pulse propagation. Journal of Computational Physics 1997;138(2):480 – 500. doi:DOI: 10.1006/jcph.1997.5827.

[9] Luebbers R, Schuster J, Wu K. Full wave propagation model based on moving window FDTD. In: Military Communications Conference, 2003. MILCOM 2003. IEEE; vol. 2. 2003, p. 1397 –401.

[10] MAFIA a multi-purpose system designed to solve electromagnetic problems. 2010. Computer Simulation Technology; URL `http://www.cst.com/Content/Products/MAFIA/Overview.aspx`.

[11] Sun DK, Lee JF, Cendes Z. Construction of nearly orthogonal nedelec bases for rapid convergence with multilevel preconditioned solvers. SIAM J SCI COMPUT 2001;23(4):1053–1076.

[12] Newmark NM. A method of computation for structural dynamics. Journal of Engineering Mechanics Division 1959;85:67–94.

[13] Schloegel K, Karypis G, Kumar V. Parallel static and dynamic multi-constraint graph partitioning. Concurrency and Computation: Practice and Experience 2002;14(3):219–40.

[14] Ruprecht D, Muller H. A scheme for edge-based adaptive tetrahedron subdivision. In: Hege HC, Polthier K, editors. Mathematical Visualization. Heidelberg: Springer Verlag; 1998, p. 61–70. URL `http://citeseer.ist.psu.edu/223528.html`.

[15] Lee LQ, Candel A, Kabel A, Li Z. On projecting discretized electromagnetic fields with unstructured grids. Tech. Rep.; SLAC National Accelerator Laboratory; 2008. URL `http://www.slac.stanford.edu/pubs/slacpubs/13250/slac-pub-13333.pdf`.

[16] The fastest supercomputer in the Top 500 list in November 2009. 2009. URL `http://www.nccs.gov/computing-resources/jaguar/`.