

# SPECTRAL-ELEMENT DISCONTINUOUS GALERKIN (SEDG) SIMULATIONS WITH A MOVING WINDOW ALGORITHM FOR WAKEFIELD CALCULATIONS

Misun Min,\* Paul F. Fischer,

Mathematics and Computer Science, Argonne National Laboratory, Argonne, IL 60439, USA

## Abstract

We developed a moving window algorithm for wakefield calculations for our SEDG time-domain electromagnetic code NekCEM. When the domain of interest is around a moving bunch within a certain distance, one does not need to carry out full domain simulations. The moving window approach is a natural choice for reducing the computational cost of the conventional low-order methods such as the finite-difference time-domain method [6]. However, there have not been studies on high-order methods, especially the SEDG method, based on the moving window approach. We implemented a 3D moving window option for wakefield calculations on various conducting cavities including a 9-cell TESLA cavity. We demonstrate the performance of the SEDG simulations on moving window meshes.

## INTRODUCTION

The Argonne-developed software NekCEM [5] is an electromagnetic solver featuring accurate and efficient computations with high performance in parallel. It uses the spectral-element discontinuous Galerkin (SEDG) method based on body-fitted spectral-element hexahedral meshes. We have demonstrated NekCEM on full-domain simulations for wakefield and wakepotential calculations on various accelerator components [1, 2]. Here we discuss the extension of our SEDG code for wakefield computations based on a moving window algorithm implemented in NekCEM. Computational results are demonstrated for tube and TESLA cavities.

## FORMULATIONS

This section addresses the governing equations, weak formulations for SEDG scheme, numerical discretizations, and initial conditions.

### Maxwell's Equations

We consider the Maxwell equations as the governing equation:

$$\mu \frac{\partial \mathbf{H}}{\partial t} = -\nabla \times \mathbf{E}, \quad \epsilon \frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{H} - \mathbf{J} \quad (1)$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon}, \quad \nabla \cdot \mathbf{H} = 0, \quad (2)$$

where  $\mathbf{E} = (E_x, E_y, E_z)^T$ ,  $\mathbf{H} = (H_x, H_y, H_z)^T$ , and  $\mathbf{J}$  represent the electric, magnetic, and current fields and  $\mu$ ,  $\epsilon$ , and  $\rho$  denote the free space permittivity, free space permeability, and space charge, respectively. The current source  $\mathbf{J} = (0, 0, J_z)^T$  and space charge  $\rho$  are defined for an on-axis Gaussian beam moving in the  $z$ -direction:

$$J_z = c\rho \quad \text{and} \quad \rho = \rho(r)\rho(z - ct), \quad (3)$$

where  $c$  is the speed of light and  $\rho$  is defined by the Gaussian distributions as

$$\rho(r) = \frac{1}{\sigma_r \sqrt{2\pi}} \exp\left(-\frac{r^2}{2\sigma_r^2}\right) \quad \text{for} \quad r^2 = x^2 + y^2, \quad (4)$$

and

$$\rho(z - ct) = \frac{1}{\sigma_z \sqrt{2\pi}} \exp\left(-\frac{(z - ct)^2}{2\sigma_z^2}\right). \quad (5)$$

We can rewrite equations (1) as

$$\mathbf{Q} \frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{S} \quad (6)$$

by defining the field vector  $\mathbf{q}$  and the flux  $\mathbf{F}(\mathbf{q})$

$$\mathbf{q} = \begin{bmatrix} \mathbf{H} \\ \mathbf{E} \end{bmatrix}, \quad \mathbf{F}(\mathbf{q}) = \begin{bmatrix} e_i \times \mathbf{E} \\ -e_i \times \mathbf{H} \end{bmatrix}, \quad (7)$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{J} \end{bmatrix}, \quad \text{and} \quad \mathbf{Q} = \text{diag}(\mu, \mu, \mu, \epsilon, \epsilon, \epsilon). \quad (8)$$

Now we consider our computational domain  $\Omega$ . We decompose the domain  $\Omega$  into nonoverlapping hexahedral elements  $\Omega^e$  such that  $\Omega = \cup_{e=1}^E \Omega^e$ . We formulate a weak form on each  $\Omega^e$  by multiplying a test function  $\phi$  to equation (6) and integrating it by parts as follows:

$$\left( \mathbf{Q} \frac{\partial \mathbf{q}}{\partial t} - \mathbf{F}(\mathbf{q}) \nabla \cdot \phi - \mathbf{S}, \phi \right)_{\Omega^e} = (-\hat{n} \cdot \mathbf{F}, \phi)_{\partial \Omega^e}. \quad (9)$$

We introduce a numerical flux  $\mathbf{F}^*$  by replacing the flux term  $\mathbf{F}$  on the right-hand side of the equation (9). Integrating by parts again gives

$$\left( \mathbf{Q} \frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) - \mathbf{S}, \phi \right)_{\Omega^e} = (\hat{n} \cdot [\mathbf{F} - \mathbf{F}^*], \phi)_{\partial \Omega^e}, \quad (10)$$

where the numerical fluxes  $\mathbf{F}^*$  are chosen as defined in [4].

**Beam Dynamics and Electromagnetic Fields**

**D05 - Code Developments and Simulation Techniques**

\* mmin@mcs.anl.gov

### Numerical Scheme

We seek the local solutions  $\mathbf{q}_N$

$$\left( Q \frac{\partial \mathbf{q}_N}{\partial t} + \nabla \cdot \mathbf{F}_N - \mathbf{S}_N, \phi \right)_{\Omega^e} \quad (11)$$

$$= (\hat{n} \cdot [\mathbf{F}_N - \mathbf{F}^*], \phi)_{\partial\Omega^e}, \quad (12)$$

where the local solution  $\mathbf{q}_N$  on each  $\Omega^e$  is defined as

$$\mathbf{q}_N(\mathbf{x}, t) = \sum_{j=0}^N q_j(t) L_j(\mathbf{x}). \quad (13)$$

assuming that  $\mathbf{F}_N$  and  $\mathbf{S}_N$  are polynomial representations of the flux and the source, respectively. Here  $q_j(t)$  is the solution at  $\mathcal{N}$  grid points  $\mathbf{x}_j$  on  $\Omega^e$ , and  $L_j(\mathbf{x})$  is the three-dimensional Legendre Lagrange interpolation polynomial of degree  $N$  associated with the  $\mathcal{N} = (N + 1)^3$  nodes [3]. The local discontinuous test function is chosen to be  $\phi = L_i(\mathbf{x})$ . Gauss-Lobatto quadrature is applied for the spatial integration. For the time advancing, we use the fourth-order Runge-Kutta method.

### Initial Conditions

To provide the electromagnetic fields at an initial time step  $t = 0$  in the presence of the Gaussian beam, we solve the Poisson equation in two dimensions at the cross section of the initial beam position  $z = z_0$ ,

$$\nabla^2 \Phi(r) = -\frac{\rho(r)\rho(z - z_0 - ct)}{\epsilon}. \quad (14)$$

In fact, the two-dimensional electric fields are immediately available at the cross section of all the values on  $z$ , which defines the initial conditions in three dimensions for the ultra-relativistic limit.

## ALGORITHMS

Here we discuss our moving window algorithm based on the spectral-element meshes. In the  $z$ -direction we restrict each element to be of equal length  $l$ . We denote the length of a moving window frame as  $L$ . We set  $M$  as the number of elements in  $z$  such that  $L = M * l$ . We choose the time step to be

$$\Delta t = l / (cN * mstep) < CFL \min(\Delta \mathbf{x}) / c, \quad (15)$$

with a proper number for  $mstep$ . In order to define the Gaussian beam properly, the moving frame size has to be

$$10 * \sigma_z < L. \quad (16)$$

### Moving Window Meshing

Assume that we have full-mesh information for the tube or TESLA cavity. As the beam moves in the positive  $z$ -direction, we update the field information as full-domain simulations in the first frame. Once the beam moves a

### Beam Dynamics and Electromagnetic Fields

### D05 - Code Developments and Simulation Techniques

distance of  $l = N * mstep * (c\Delta t)$ , we construct a new mesh for the next frame by reading partial information of the next frame from the existing full mesh data. This approach makes our algorithm very efficient. Figure 1 shows moving window simulations for a tube and the magnitude of the electric field distribution in time with three moving window frames being used. For the field information we copy the electromagnetic fields only in the middle zone  $\Gamma$ , discard the data in the back zone of the beam  $\Gamma_B$  (dashed line box in white), and add field values as zeros for the front zone  $\Gamma_F$  (dashed line box in black).

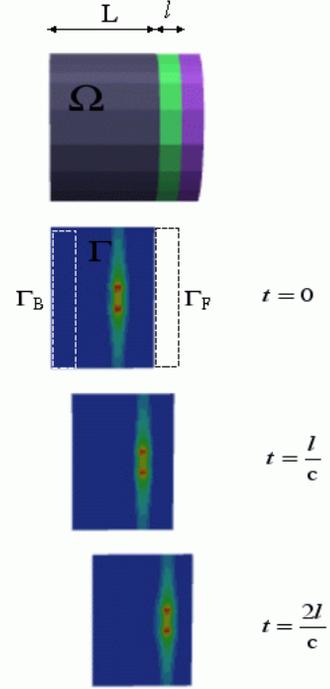


Figure 1: Moving windows for tube and the magnitude of the electric field at time  $t = 0, l/c, \text{ and } 2l/c$ .

### Boundary Conditions

Perfectly electric boundary conditions are applied for the transverse direction. In the longitudinal direction  $z$ , boundary conditions need not be specified. In the front of the beam moving at the speed of light, there is no field defined. The fields in the back of the beam do not catch any field interaction that will happen in the next moving frame. Thus one does not need to assign any boundary conditions in the  $z$ -direction for each moving frame.

## COMPUTATIONAL RESULTS

Here we demonstrate the efficiency of our moving algorithm. The CPU time depends on the length of the moving frame.

## Performance

To demonstrate the performance of the moving window algorithm, we compute the case with wakefield calculations for a tube mesh with the total number of the elements  $E = 900$  with 9 elements in transverse direction and 100 elements in longitudinal direction. Computations are performed on an Intel XEON processor with 2.40 GHz. Figure 2 plots CPU time vs. the number of elements  $M$  in a moving window frame with increasing  $M = 5, 15, 45, 100$ . The full size of the domain in  $z$  is  $M = 100$ . The figure shows that CPU time decreases linearly depending on the length of the moving frame size. The polynomial order is fixed with  $N = 6$ . This result implies that one can obtain better efficiency and accuracy with the high-order method with the moving window option presented in this paper.

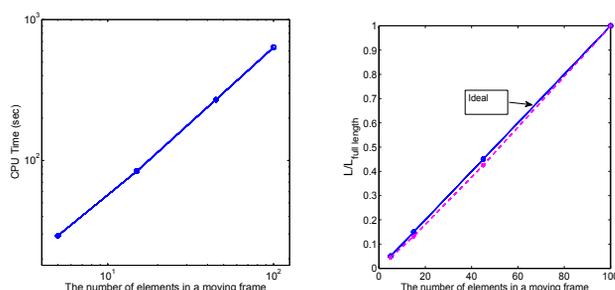


Figure 2: CPU time vs. moving frame length in log scale, showing that CPU time increases linearly depending on the length of the moving frame. The number of elements in a moving frame is  $M = 5, 15, 45, 100$  (left); the ratio of the moving window frame size over the full length of the domain in dashed line, showing that the ratio is close to the ideal case in solid line (right).

## Discussion

Figure 3 shows TESLA cavity moving window meshes in time. Because of the restriction on the  $z$ -direction that the element sizes must be equal, our TESLA mesh is highly deformed in the  $z$ -direction, with sharp angles around the transverse boundaries. We plan to improve the mesh quality for TESLA cavity in future work. We will parallelize the moving window and carry out performance studies for large scale simulations.

## CONCLUSIONS

We have implemented a moving window algorithm in our SEDG electromagnetic solver NekCEM. Equipped with the moving window option, high-order methods are comparable to existing lower-order wakefield calculation codes. Our moving window algorithm can reduce the CPU time by a factor of the moving frame size over the full length of the domain. Once parallelized, the moving window algorithm with high-order simulations will be a

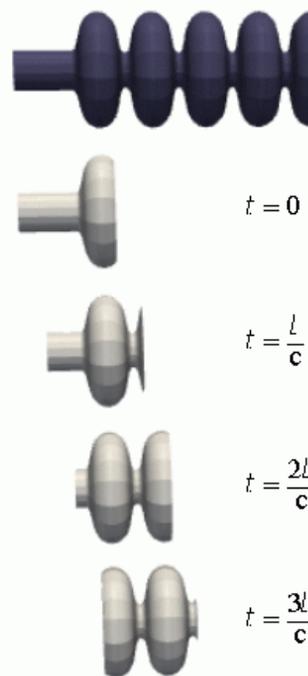


Figure 3: Moving windows for TESLA cavity at time  $t = 0, l/c, 2l/c, \text{ and } 3l/c$ .

promising tool to track a 1 ps beam moving through meter-scale cavities.

## ACKNOWLEDGMENT

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.

## REFERENCES

- [1] M.S. Min, P.F. Fischer, Y.C. Chae, "Wake fields for TESLA cavity structures: Spectral element discontinuous Galerkin simulations," Proc. of SRF07, TUP34, 2007.
- [2] M.S. Min, P.F. Fischer, Y.C. Chae, "Spectral-element discontinuous Galerkin simulations for bunched beam in accelerating structures," Proc. of PAC07, pp. 3432-3434, 2007.
- [3] M.O. Deville, P.F. Fischer, and E.H. Mund, "High Order Methods for Incompressible Fluid Flow," Cambridge University Press (2002).
- [4] J.S. Hesthaven and T. Warburton, "Nodal High Order Methods on Unstructured Grids - I. Time-Domain Solution of Maxwell's Equations," J. Comp. Physics, 101 (2002), p. 186.
- [5] M.S. Min and P.F. Fischer, "NekCEM," Mathematics and Computer Science Division, Argonne National Laboratory. <https://svn.mcs.anl.gov/repos/NEKCEM>.
- [6] A. Taflov and S.C. Hagness, "Computational Electromagnetics: The Finite-Difference Time Domain Method," Artech House, 3rd ed. (2005).