Journal of Computational Physics 228 (2009) 7902–7916

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Application of Dey–Mittra conformal boundary algorithm to 3D electromagnetic modeling ☆

C. Nieter [a,*], John R. Cary [a,b], Gregory R. Werner [b], David N. Smithe [a], Peter H. Stoltz [a]

[a] Tech-X Corporation, Beam-Plasma Interactions Group, 5621 Araphoe Avenue, Suite A, Boulder, CO 80303, United States
[b] Center for Integrated Plasma Studies and Department of Physics, University of Colorado, Boulder, CO 80309, United States

ARTICLE INFO

Article history:
Received 12 December 2007
Received in revised form 25 June 2009
Accepted 16 July 2009
Available online 3 August 2009

Keywords:
Electromagnetic
FDTD
Conformal
Boundaries

ABSTRACT

The Dey–Mittra conformal boundary conditions have been implemented for the finite-difference time-domain (FDTD) electromagnetic solver of the VORPAL plasma simulation framework and studied in the context of three-dimensional, large-scale computations. The maximum stable time step when using these boundary conditions can be arbitrarily small, due to the presence of small fractional cells inside the vacuum region. Use of the Gershgorin Circle theorem allows the determination of a rigorous criterion for exclusion of small cells in order to have numerical stability for particular values of the ratio $f_{DM} \equiv \Delta t/\Delta t_{CFL}$ of the time step to the Courant–Friedrichs–Lewy value for the infinite system. Application to a spherical cavity shows that these boundary conditions allow computation of frequencies with second-order error for sufficiently small f_{DM}. However, for sufficiently fine resolution, dependent on f_{DM}, the error becomes first order, just like the error for stair-step boundary conditions. Nevertheless, provided one does use a sufficiently small value of f_{DM}, one can obtain third-order accuracy through Richardson extrapolation. Computations for the TESLA superconducting RF cavity design compare favorably with experimental measurements.

1. Introduction

The simulation of electromagnetics in complex structures has traditionally been done with finite element algorithms using unstructured meshes. There are many specific situations where a finite difference representation of Maxwell's equations is preferred over the finite element representation. However, the application of finite difference algorithms to these problems has been limited since the regular Cartesian meshes normally used with finite difference do not represent structures with complex, curved geometries accurately unless a very fine grid resolution is used. Recent developments in boundary algorithms have opened to door to accurate simulations of such structures with finite difference methods. Due to the challenges associated with the development of three-dimensional algorithms as well as the expense of running large three-dimensional simulations the testing of these algorithms has generally been limited to two dimensional situations.

Many complex three-dimensional structures are used in modern electromagnetic applications in science and engineering. Examples include magnetrons used to generate high-power microwaves and radio frequency cavities used in particle accelerators. The geometry of these devices has evolved to very complex structures with a variety of curved surfaces making the representation of such devices with a regular Cartesian mesh difficult. Although these cavities do often have some degree of

☆ This work was supported by the US Department of Energy under Grants DE-FG02-04ER41317, DE-FG02-05ER84172, and by the Air Force under Grants FA9451-06-M-0095.

* Corresponding author. Tel.: +1 303 444 2582; fax: +1 303 448 7756.
E-mail addresses: nieter@txcorp.com, chet.nieter@gmail.com (C. Nieter).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.025

symmetry associated with them, there are usually coupling structures which break this symmetry and are often the location of the interesting physics.

The most common method of modeling these structures with a finite difference code is to stair-step the boundary where the structure curves. This results in very poor resolution of the geometry unless a very fine grid is used which often results in an unacceptable increase in computational cost. Recent developments in boundary algorithms modify the finite difference update at the boundary to take into account the curved surfaces. These are often referred to as cut cell methods as the finite difference stencil will take into account how the boundary cuts across the grid cells.

Although it can be argued that finite element algorithms on an unstructured mesh are better suited to dealing with such complex geometries, there are many cases where it is appropriate to use finite difference methods. The finite difference approach provides a critical advantage in terms of equivalency between the discrete Maxwell and continuity equations and the exact continuous equations, which describe the real world. For the typical choice of the Buneman [1] current allocation and the Yee [2] finite differencing algorithm, there are difference-equation analogues to all important vector differential identities satisfied in the real world. For example, there is a difference-equation version of the Poynting theorem that ensures energy conservation to machine precision, for the electromagnetic fields (in absence of particles). Since most of these identities are not well defined at boundaries in the continuum the addition of boundary methods to FDTD simulations does not impact this advantage since they still apply away from the boundaries.

The physics of the devices mentioned above often include the presence of charged particles interacting with the fields and cavity structures. The use of Particle-In-Cell (PIC) methods remains the most popular way of modeling the presence of charged particles interacting with electromagnetic fields. In simulations where a full Maxwell treatment of the fields is required and the charged particles are modeled with PIC, finite difference methods are better suited since depositing charge and current from the particles to the grid and the interpolation of the fields back to the particles positions are well understood when using a structured, Cartesian mesh. There are complications with particle dynamics in the cut cells including how properly to do the field interpolation and how to prevent the creation of unphysical image charges, but there are methods to deal with these issues.

2. Background

The Yee [2] finite difference algorithm simulates electromagnetic fields on a regular grid. The Yee algorithm achieves second-order error by centering all finite differences in space and time; this requires offsetting the components of the electric and magnetic fields as shown in Fig. 1, as well as offsetting the electric and magnetic fields by one-half time step. For example, the magnetic field is advanced in time by the following finite difference version of $\partial B_z/\partial t = -(\nabla \times \mathbf{E})_z$:

$$B^{n+1/2}_{z;i+1/2,j+1/2,k} = B^{n-1/2}_{z;i+1/2,j+1/2,k} + \frac{\Delta t}{A_{xy}} \times \left(E^{n}_{x;i+1/2,j,k} \cdot l_x - E^{n}_{x;i+1/2,j+1,k} \cdot l_x + E^{n}_{y;i+1,j+1/2,k} \cdot l_y - E^{n}_{y;i,j+1/2,k} \cdot l_y \right), \tag{1}$$

where $B^{n+1/2}_{z;i+1/2,j+1/2,k}$ is the z-component of the magnetic field in cell (i,j,k) at time $t = (n+1/2)\Delta t$; l_x and l_y are the lengths of the cell edges on which E_x and E_y are located, and $A_{xy} = l_x l_y$ is the area of the face centered at B_z.

All field components in Eq. (1) are located on the same cell face, with B_z at its center and E_x and E_y on the edges. This algorithm is appropriate when no conducting boundary cuts through the face.

When the conducting boundary cuts through a cell face, we can approximate the boundary with a nearby boundary that never cuts through cells – i.e. stairstepping – and use the Yee algorithm, setting the electric field to zero on cell edges that border the conducting boundary. Stairstepping a boundary introduces global first-order error (e.g. the frequency of a simulated cavity would have an error scaling as the cell-length).

Second-order error can be attained using the Dey–Mittra [3] algorithm, which can simulate partial faces. The Dey–Mittra algorithm advances the electric field in the same manner as the Yee algorithm. The only difference is the electric field is set to zero on edges totally contained within the conductor. The Yee algorithm is also used to advance the magnetic field on cell faces uncut by the conducting boundary. On cell faces cut by the boundary, however, the Dey–Mittra algorithm adjusts the edge lengths and the face areas to reflect the lengths and areas not inside the conductor: for example,

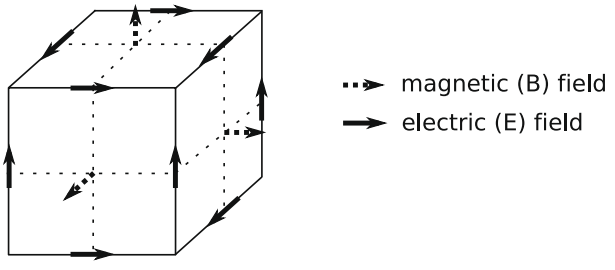

Fig. 1. Field components in the Yee mesh.

$$B^{n+1/2}_{z;i+1/2,j+1/2,k} = B^{n-1/2}_{z;i+1/2,j+1/2,k} + \frac{\Delta t}{A_{xy;i+1/2,j+1/2,k}} \times \left(E^n_{x;i+1/2,j,k} \cdot l_{x;i+1/2,j,k} - E^n_{x;i+1/2,j+1,k} \cdot l_{x;i+1/2,j+1,k} + E^n_{y;i+1,j+1/2,k} \cdot l_{y;i+1,j+1/2,k} - E^n_{y;i,j+1/2,k} \cdot l_{y;i,j+1/2,k} \right). \quad (2)$$

Here, $A_{xy;i+1/2,j+1/2,k}$ is the area of the face (where $B_{z;i+1/2,j+1/2,k}$ is located) outside the conductor, and similarly for the edge lengths.

Eq. (2) essentially advances B_z by taking the line integral of $\mathbf{E}$ around the area of the cell face, excluding the conductor; of course $\mathbf{E} \cdot dl$ is zero along the conductor, leaving only the line integral of E around the parts of the edges outside the conductor. When advancing the magnetic field, the Yee algorithm uses uniform l_i and A_{ij}, whereas the Dey–Mittra algorithm adjusts the lengths and areas from cell to cell to allow simulation of partial cells.

The Dey–Mittra algorithm can lead to instability when simulating cells that are mostly contained within the conductor. Intuitively, one can think of this instability being related to the Courant–Friedrichs–Lewy condition: roughly, light must not cross more than one cell during a single time step to avoid instability. When the Dey–Mittra scheme includes very small cells, a very small time step must be used.

To avoid infinitesimally small time steps using the Dey–Mittra algorithm, one must ignore cell faces with only small area fractions outside the conductor, effectively stairstepping over such faces. By examing the Courant stability of the Dey–Mittra algorithm, we will determine a condition for deciding when a face must be ignored, given a chosen time step. Before we examine Dey–Mittra stability, we will review the stability criterion for the time step using the Yee algorithm.

3. The maximum stable time step for the Yee algorithm

The maximum stable time step for the Yee algorithm is the well-known Courant–Friedrichs–Lewy time step; we derive it using Gershgorin's circle theorem. The Maxwell wave equations can be represented as matrix-vector operations, where $\mathbf{e}$ is a column vector containing all electric field components at all locations on the grid, and similarly $\mathbf{b}$ is the magnetic field. Maxwell's equations, discretized spatially, can be written:

$$\frac{d\mathbf{b}}{dt} = -\mathbf{C} \cdot \mathbf{e}, \quad \frac{d\mathbf{e}}{dt} = \mathbf{C}^\dagger \cdot \mathbf{b},$$
$$\frac{d^2\mathbf{b}}{dt^2} = -\mathbf{C} \cdot \mathbf{C}^\dagger \cdot \mathbf{b} \equiv -\mathbf{D} \cdot \mathbf{b}, \quad (3)$$

where matrix operators $\mathbf{C}$ and $\mathbf{C}^\dagger$ are adjoints of each other, and $\mathbf{D}$ is self-adjoint and positive semi-definite; $\mathbf{C}$ is a forward-curl operator, $\mathbf{C}^\dagger$ is a backward curl operator, and $\mathbf{D}$ is basically the finite difference Laplacian operator. Diagonalizing $\mathbf{D}$ reduces the problem to a collection of simple harmonic oscillators (the eigenmodes of $\mathbf{D}$) with frequencies that are the square roots of the eigenvalues of $\mathbf{D}$.

The time discretization of a harmonic oscillator can lead to an unphysical instability if the time step is too large compared to the oscillation period. Stability for leap-frog time-integration of a simple harmonic oscillator with frequency ω requires a time step satisfying

$$\omega \Delta t < 2. \quad (4)$$

Therefore, the maximum stable time step for integrating Eq. (3) is

$$\Delta t_{\max} = 2/\omega_{\max}, \quad (5)$$

where $\omega_{\max}$ is the maximum frequency, i.e. the square root of the maximum eigenvalue of the operator $\mathbf{D}$.

The maximum stable time step depends on the maximum eigenvalue of $\mathbf{D}$. We can place an upper bound on the eigenvalues of $\mathbf{D}$ using the Gershgorin Circle theorem [4], which tells us that the eigenvalues of $\mathbf{D}$ must satisfy

$$\omega^2/c^2 \leqslant \max_i \left(\sum_j |D_{ij}| \right), \quad (6)$$

where D_{ij} are the matrix elements of $\mathbf{D}$, and ω^2/c^2 are the eigenvalues of $\mathbf{D}$ – as we have written Eq. (3) – and c is the speed of light.

In the Yee algorithm, $\mathbf{D}$ is basically the finite difference Laplacian operator $(-\nabla^2)$, with

$$D_{ii} = \frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2}{\Delta z^2}. \quad (7)$$

Besides the diagonal, each column of $\mathbf{D}$ contains only six other non-zero elements: $-1/\Delta x^2$ twice, $-1/\Delta y^2$ twice, and $-1/\Delta z^2$ twice. Applying Gershgorin's circle theorem we see that eigenvalues of $\mathbf{D}$ must satisfy:

$$\frac{\omega^2}{c^2} \leqslant \frac{4}{\Delta x^2} + \frac{4}{\Delta y^2} + \frac{4}{\Delta z^2} \quad (8)$$

with each column having $2d$ other non-zero elements which are all one. From this we can conclude that for stability,

$$\Delta t \leqslant \frac{2}{\omega_{\max}} \leqslant \frac{1}{c\sqrt{\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}}}. \tag{9}$$

In fact, this is exactly the Courant–Friedrichs–Lewy condition.

For completeness, we point out the importance of having $\omega^2/c^2 \geqslant 0$, which is guaranteed because $\mathbf{D}$ is positive semi-definite (as is any matrix of the form $\mathbf{CC}^\dagger$). If ω^2/c^2 could be negative or complex, then the frequency ω could have an imaginary part, corresponding to exponentially growing and decaying modes that violate the conservation of energy.

4. Stability for the Dey–Mittra algorithm

To avoid instability with the Dey–Mittra scheme, the time step must be reduced below the Courant time step used for the Yee algorithm, because the Dey–Mittra algorithm allows smaller cells. Explicitly, we can write the Dey–Mittra algorithm as

$$\frac{d^2\mathbf{b}}{dt^2} = -\mathbf{A}^{-1}\cdot\mathbf{C}\cdot\mathbf{L}\cdot\mathbf{C}^\dagger\cdot\mathbf{b} \equiv -\mathbf{D}'\mathbf{b}, \tag{10}$$

where $\mathbf{C}$ is the same as in the Yee algorithm, Eq. (3), and $\mathbf{A}$ is a diagonal matrix containing the area fractions of each face outside the conductor ($\mathbf{A}$ is diagonal because each face corresponds a unique element of $\mathbf{b}$), and L is a diagonal matrix containing the length fractions for each edge (each edge corresponds to an element of $\mathbf{e}$). When a cell has a small fractional area, $\mathbf{A}^{-1}$ has a large diagonal element, and therefore $\mathbf{D}'$ has a large eigenvalue, which forces the time step to be small (Eq. (5)).

If one cell face in the simulation has nearly vanishing area, a nearly infinitesimal time step must be used. Therefore, an efficient simulation neglects (stairsteps) such small (fractions of) cells to allow a larger time step. Using the Gershgorin circle theorem, we can determine which cut faces can be neglected to guarantee stability with a given time step. Whether a cell face should be neglected depends not just on the fractional area of the face, but also on the fractional edge lengths.

We define the desired time step to be $f_{DM}\Delta t_{\mathrm{CFL}}$ where Δt_{CFL} is the Courant–Friedrichs–Lewy time step, given in Eq. (9).

There are three possible cut-cell types whose stability must be checked. The first is when two adjacent edges on a cell face are cut where the resulting triangle is outside the boundaries. In this case the face is stable for a full Courant time step and nothing else needs to be done. The second case is again when two adjacent edges are cut but now the triangle is inside the boundary. The third case is when two opposite edges are cut created a trapezoid. These three cases are shown in Fig. 2. We consider each case in turn by writing out the magnetic field update as a line integral around the face,

$$\frac{d\Phi_B}{dt} = \frac{A_{xy}dB_z}{dt} = -\oint \mathbf{E}\cdot\mathbf{dl} \simeq -\sum \mathbf{E}_i\cdot\mathbf{l}_i, \tag{11}$$

where Φ_B is the magnetic flux though the face, A_{xy} is the area of the cut face, and l_i is the length of the cut cell sides. Taking the time derivative of Eq. (11) we arrive at

$$\frac{A_{xy}d^2B_z}{dt^2} = -\sum \frac{d\mathbf{E}_i}{dt}\cdot\mathbf{l}_i, \tag{12}$$

By replacing the dE_i/dt terms with the finite difference representation of Ampere's Law we can collect all the matrix terms and apply Eq. (6) to determine the stability condition for the cell. Since l_i, A_{xy}, and the number of terms in the sum in Eq. (12) will vary depending on how the cell is cut, each boundary cell will have a different stability condition.

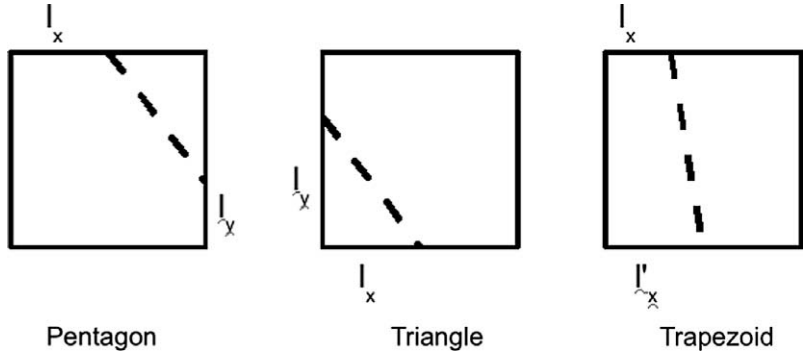

Fig. 2. The various ways a conformal boundary can cut across the face of a Yee cell. The area to the right of the cut (dashed line) in these examples is assumed to be metal.

As an example of how this is done we consider the case where two adjacent sides are cut by the boundaries and the triangle formed by this cut is the part of the cell that is inside the simulation domain. This leaves us with only two terms in the sum of Eq. (12),

$$\frac{A_{xy}d^2B_z}{dt^2} = -\frac{dE_x}{dt}l_x + \frac{dE_y}{dt}l_y. \tag{13}$$

Replacing the time derivatives of the electric field using the finite difference representation of Ampere's Law we arrive the following equation,

$$\frac{d^2B_z}{dt^2} = c^2\left[\frac{l_x}{A_{xy}}\left(\frac{\Delta B_y}{\Delta z} - \frac{\Delta B_z}{\Delta y}\right) + \frac{l_y}{A_{xy}}\left(\frac{\Delta B_x}{\Delta z} - \frac{\Delta B_z}{\Delta x}\right)\right]. \tag{14}$$

We can identify the matrix terms for the cell from this equation. By applying the Gershgorin theorem we do not need to know the structure of the matrix just the value of all the terms for the cell. By combining the stability condition from Eq. (4) and the constraints on eigenvalues of Eq. (14) from the Gershgorin theorem we find that if the following equation is satisfied then the Dey–Mittra update in the cell in question will be stable,

$$\frac{\frac{1}{f_x\Delta x^2} + \frac{1}{f_y\Delta y^2} + \frac{1}{f_x\Delta x\Delta z} + \frac{1}{f_y\Delta y\Delta z}}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} < \frac{\Delta t_{CFL}^2}{\Delta t^2} = \frac{1}{(f_{DM})^2}, \tag{15}$$

where $f_x = l_x/\Delta x, f_y = l_y/\Delta y$ so $A_{xy} = \frac{1}{2}f_x f_y \Delta x\Delta y$ and Δt_{CFL} is the Courant time step for a normal cell.

Taking limit of $\Delta z \to \infty$ to get the stability condition,

$$\frac{\frac{1}{f_x\Delta x^2} + \frac{1}{f_y\Delta y^2}}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} < \frac{\Delta t_{CFL}^2}{\Delta t^2} = \frac{1}{(f_{DM})^2}. \tag{16}$$

Following a similar procedure for faces cut by the boundary on opposite edges (the x-edges in this case) gives the following stability condition in 3D,

$$\frac{\frac{1}{2a\Delta x^2} + \frac{1}{2a\Delta x\Delta z} + \frac{1}{\Delta y^2} + \frac{1}{\Delta y\Delta z}}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} < \frac{\Delta t_{CFL}^2}{\Delta t^2} = \frac{1}{(f_{DM})^2}, \tag{17}$$

where $a = A_{xy}/\Delta x\Delta y$. There are now two possibilities in 2D. The first is when the unsimulated direction is perpendicular to the face. The stability condition for this case is found by taking the limit $\Delta z \to \infty$,

$$\frac{\frac{1}{2a\Delta x^2} + \frac{1}{\Delta y^2}}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} < \frac{\Delta t_{CFL}^2}{\Delta t^2} = \frac{1}{(f_{DM})^2}. \tag{18}$$

The second possibility is when the uncut edges of the face are in the unsimulated direction. In this case the stability condition is found by taking the limit $\Delta y \to \infty$,

$$\frac{\frac{1}{2a\Delta x^2} + \frac{1}{2a\Delta x\Delta z}}{\frac{1}{\Delta x^2} + \frac{1}{\Delta z^2}} < \frac{\Delta t_{CFL}^2}{\Delta t^2} = \frac{1}{(f_{DM})^2}. \tag{19}$$

With a time step Δt satisfying the above condition, the simulation is guaranteed to be stable, but Δt is not necessarily the maximum stable time step.

5. Application to frequency calculations

To extract the mode frequency of a cavity from a simulation, we used a common method of exciting modes of known frequencies. At the beginning of the simulation a current source whose spatial profile approximates the profile of the known mode is initially applied to the simulation. The current source will have a gaussian envelope in time and with a center frequency that matches the known mode frequency. By having the envelope cover several periods of the mode, the current source will be peaked around the mode frequency in frequency space. By choosing the breadth of the gaussian we can insure the peak is narrow enough so only the mode of interest is excited. Once the current source is shut off the cavity will continue to oscillate at the mode frequency. The bottom plot in Fig. 3 shows the current for a typical current source used to ring up the cavity mode, with the resulting electric field in the same direction as the current source shown in the top plot of Fig. 3. This method of driving cavity can be used to study the mode structure of cavities whose modes cannot be determined analytical or it can be used to ring cavities with known mode structures to compare frequencies between the simulated cavities and the real cavities.

We use this method of driving cavity oscillations to perform convergence tests of the Dey–Mittra algorithms. We run simulations for a known mode of a specific cavity shape where we keep the physical dimensions of the cavity fixed but we increase the number of grid cells used in each direction. This gives us a series of simulations at greater and greater grid

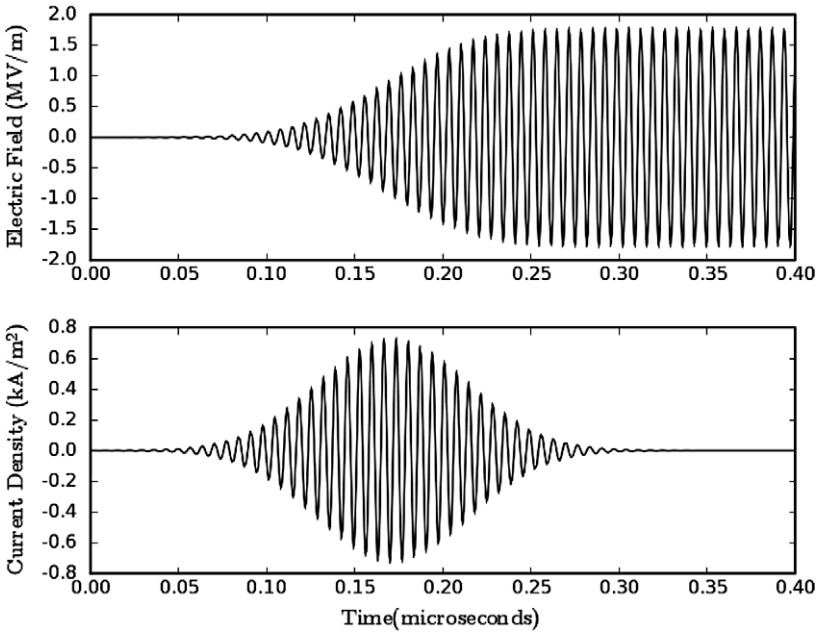

Fig. 3. Time series of the current driver (lower plot) and the resulting electric field (upper plot) for a typical single mode cavity simulation used to determine the mode frequency of the cavity.

resolution so we can determine how the simulation results converge to the physical value of the mode frequency. These simulations were run using the finite-difference time-domain electromagnetic field solver of the VORPAL plasma simulation framework. The electromagnetic solver in VORPAL has both Dey–Mittra boundaries as well as stair-step boundaries.

To extract the mode frequencies we record the field at every time step at a point in the cavity close to a spatial maximum of the mode. We then fit this time series (excluding the time while the driving current is on) to a sum of cosines where the amplitude, frequency and phase of each term is a free parameter. Since the cavity oscillation is dominated by the mode of interest only a few terms are needed, typically two or three. The term with the highest amplitude corresponds to the mode of interest so that fitted frequency is taken as the mode frequency.

The first case that we determine the frequency convergence for is the lowest mode of a spherical cavity. We consider the lowest mode of the cavity where the electric field is in one direction in the cavity. The mode has a continuous degenerancy depending on the direction that the electric field is pointed. By driving a current in the cavity we pick a direction for the field.

In Fig. 4 we plot the error in the measured cavity frequency for the convergence tests for the spherical cavity. The triangles are the measured frequencies for a series of VORPAL runs using stair-step boundaries and the circles are the measured frequencies for a series of runs using the Dey–Mittra boundaries with $f_{DM} = 0.25$. The dashed straight lines represent first and second order scaling in the grid spacing. The error for the Dey–Mittra algorithm is in general several orders of magnitude lower than stair-step boundaries at the same grid spacing. The stair step data closely follows the first order scaling line where the Dey–Mittra data follows the second order line. However, for very fine grid resolutions the Dey–Mittra results begin to show a first order tail.

The second case that was studied is a standing wave in a rectangular wave guide. The ends of the wave guide are closed forming a box cavity whose modes correspond to a standing wave of a rectangular wave guide mode. The input file for these runs is designed so the orientation of the cavity axis can be rotated with respect to the grid axis where both the azimuthal and zenith angles can be varied. This allows us to test the Dey–Mittra algorithm by having the cavity walls cut across the grid cells.

In Fig. 5 we plot the error in the measured frequency for a closed waveguide which is rotated by 60° in the $x - y$ plane and by 30° in the $x - z$ plane. Again the triangles are the measured frequencies from runs using stair-step boundaries and the circles are from runs using Dey–Mittra boundaries with $f_{DM} = 0.25$. The dashed straight lines represent first and second order scaling in the grid spacing. Again the Dey–Mittra results follow the second order line and the stair step results follow the first order line.

The spherical and box cavity provide good test cases since a closed analytical solution exists for such geometries allowing comparisons to made with known results. However, the geometries that are studied by codes such as VORPAL are considerably more complex. Superconducting radio frequency cavities (SRF) used in particle accelerators are an good example of a situation where an electromagnetic-PIC code like VORPAL which can model complex boundaries would be useful [5]. A simulation of the nine cell TESLA cavity design [6] was done to try and identify a set of experimentally known higher order modes. The TESLA cavity consists of nine elliptical superconducting cavities and it is under development for use as an

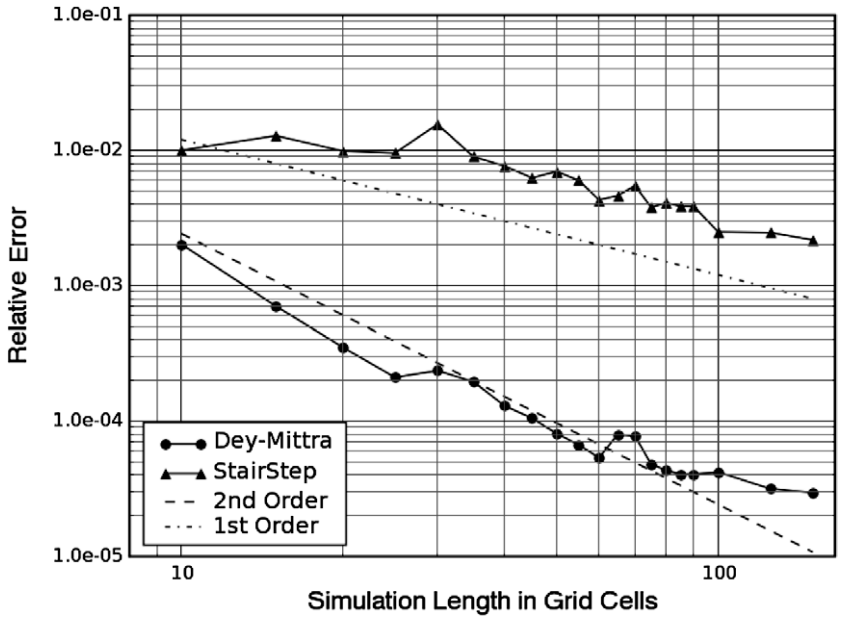

Fig. 4. Relative frequency error for the lowest mode of a spherical cavity plotted against the number of grid cells used in each dimension. The triangles are results from simulations using stair-step boundaries and the circles are results from simulations using the Dey–Mittra boundaries with $f_{DM} = 0.25$. The dashed straight lines represent first and second order convergence and are not fits to the computational data.

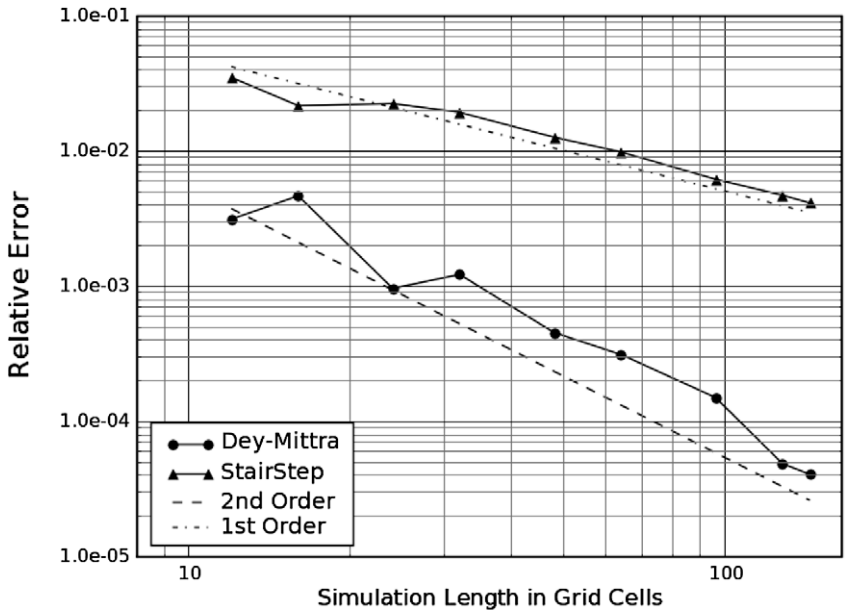

Fig. 5. Relative frequency error for the lowest mode of a closed rectangular waveguide cavity plotted against the number of grid cells used in each dimension. The triangles are results from simulations using stair-step boundaries and the circles are results from simulations using the Dey–Mittra boundaries with $f_{DM} = 0.4$. The dashed straight lines represent first and second order convergence and are not fits to the computational data.

accelerating cavity in the next generation of electron-positron colliders. The cavity modes were excited using a current pulse similar to the ones used in our convergence runs but with broader support in frequency space so multiple modes would be excited. The current source was present only in the first cavity and the electric field was then measured at every time step in the same location as the driving current. The simulation was run for a long period of time (approximately 1000 periods of the lowest TE111 mode) to provide good statistics to calculate a power spectrum. The solid line in Fig. 6 is a power spectrum calculated by a VORPAL simulation showing several modes of the cavity. The individual data points are experimental data

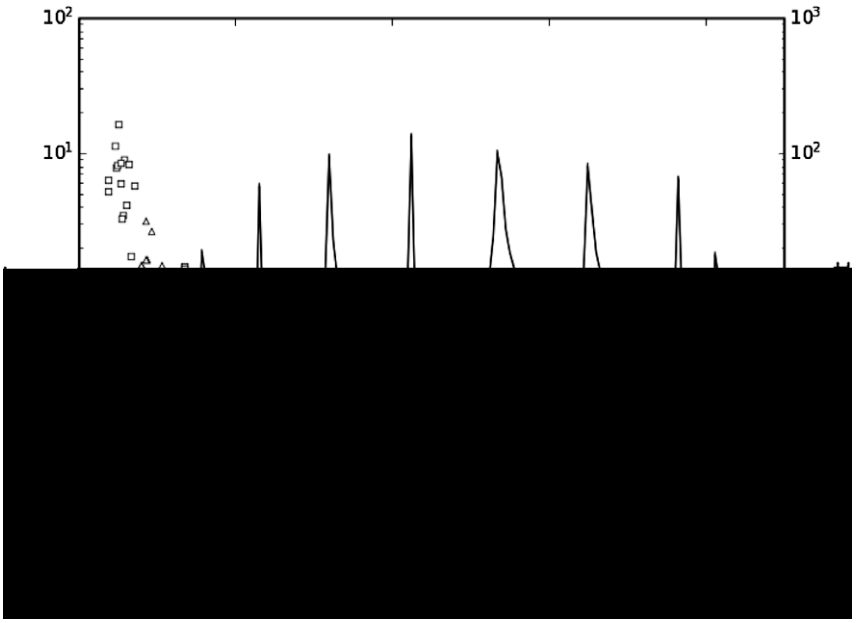

Fig. 6. Comparison of experimental measured mode frequencies of the nine cell Tesla cavity with those found using VORPAL. The solid line is a power spectrum calculated by a VORPAL simulation. The triangles and squares are experimentally measured frequencies for the TE111 modes. The symbols for the mode frequencies is alternated so the data for each mode can be distinguished. Each group of triangles or squares represents results for a single mode.

[7] for the TE111 modes (transverse electric and transverse magnetic) of the cavity. Each group of similar symbols (squares and triangles) represents multiple experimental results for the same frequency. The location of the peaks in the VORPAL power spectrum correlates with the average location of the experimental data on frequency axis for each of the TE111 modes. An additional TE111 mode frequency appears in the VORPAL data. It is possible that this could have been misidentified in the TESLA experimental data as a TM011 mode so it does not appear in the experimental data for the TE111 frequency band.

6. Transition between second- and first-order convergence

The frequency errors using the Dey–Mittra scheme for a spherical cavity (see Fig. 4) converge as the second power of the grid cell size (Δx) for large cell sizes, but for smaller cell sizes the errors decrease with first-order in Δx. This transition arises from the neglect of small fractional cell faces that allows the use of a reasonable time step (cf., Section 4). Fig. 7 shows that the transition point can be pushed to higher resolutions by using smaller f_{DM} – i.e. neglecting fewer fractional faces, at the cost of reducing the maximum stable time step. If no faces are neglected, the error will always be second-order. Since f_{DM} represents the actual fraction of the Courant time step that keeps the accepted fractional cell faces stable the cost in terms of run time is inversely proportional to f_{DM}. However, once no faces are neglected no further benefit is gained by further reduction in f_{DM}.

We have more thoroughly explored the effect of f_{DM} on the convergence of mode frequencies in a two-dimensional elliptical cavity; we considered TE modes (where the electric field is entirely in the plane) of an elliptical cavity with major axis 1 m and minor axis 0.7 m, tilted with respect to the grid axes. As f_{DM} is decreased, fewer cut faces are neglected; Fig. 8 shows the fraction of neglected faces versus f_{DM} for several different grid resolutions; empirically we find that for the elliptical cavity discussed in this section

$$\frac{\text{number of neglected faces}}{\text{number of total faces}} \approx f_{DM}^2 \frac{P\Delta x}{A} \tag{20}$$

where P is the perimeter and A the area of the ellipse.

Figs. 9–11 show the relative errors in the spatial eigenvalue (the square of the frequency) for the lowest 11 TE modes of the elliptical cavity for $f_{DM} = 0, 0.2$, and 0.5. (The TE modes are labeled as they would be in a cylindrical cavity, but with C representing a cosine-like "azimuthal" dependence, and S sine-like.) For $f_{DM} = 0$, no cut faces are neglected (the time step was chosen to be stable by trial and error) and the eigenvalue errors converge slightly better than $O(\Delta x^2)$. A relatively small number of cut faces are neglected with $f_{DM} = 0.2$, and the error looks similarly second order, though there is a hiccough when cut faces are neglected (as will be made clearer in the Section 7). For $f_{DM} = 0.5$, many cut faces are neglected, and the error is not so convincingly second-order, especially at higher resolutions, where the convergence changes from second- to first-order.

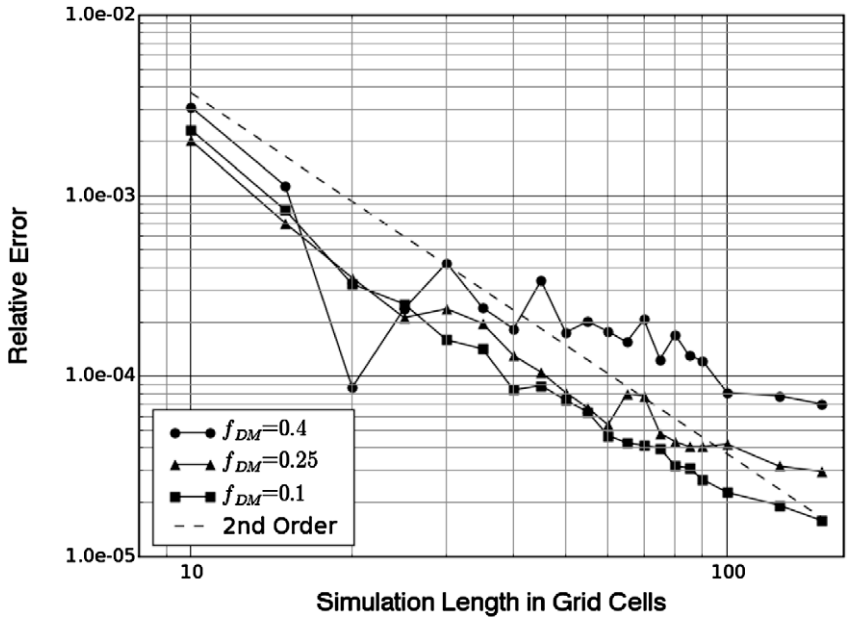

Fig. 7. Relative frequency error for the lowest mode of a spherical cavity for several different values of the f_{DM} parameter plotted against the number of grid cells used in each dimension. The circles, triangles, and squares are results for f_{DM}'s of 0.4, 0.25, and 0.1, respectively. The dashed straight line represents second order convergence and is not a fit to the computational data.

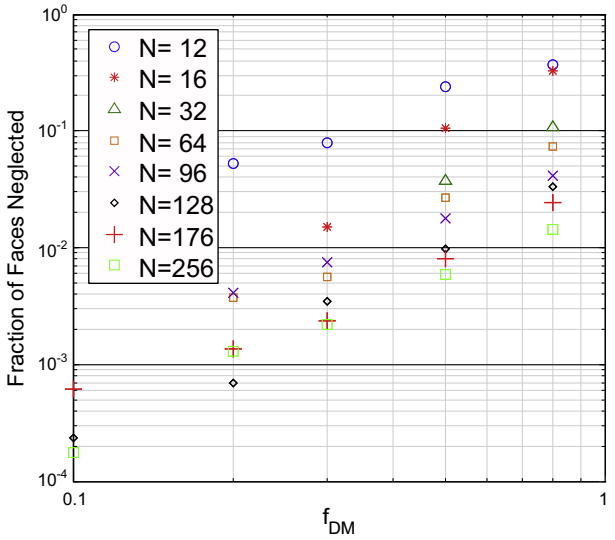

Fig. 8. The number of neglected cut faces divided by the number of total cut faces in a 2D elliptical cavity versus f_{DM}, for several different grid resolutions (N = the number of cells per cavity diameter).

7. Richardson extrapolation of results

When the mode frequency is a sufficiently smooth function of the grid cell size Δx, Richardson extrapolation can be used to find a frequency with higher order error from simulations with lower order error. For example, if we know that $f(\Delta x_1) = f_1$ and $f(\Delta x_2) = f_2$, then we can find a more accurate frequency f_0, assuming error of order n, by fitting $f(\Delta x) = f_0 + \alpha \Delta x^n$ for the best value of f_0. We found f_0 using pairs of simulations with cell sizes Δx and $\Delta x/2$, assuming $n = 2$.

When no cut faces are neglected, the Dey–Mittra method yields second-order error (Fig. 9), and can be extrapolated to third-order error, as shown by Fig. 12. However, neglecting cut faces prevents extrapolation to third-order

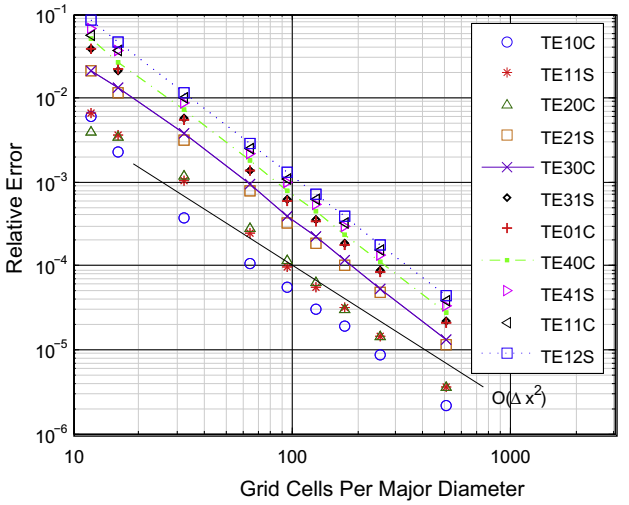

Fig. 9. The relative error in eigenvalue (frequency squared) versus grid resolution, for the lowest 11 TE modes of a 2D elliptical cavity; no cut faces are neglected.

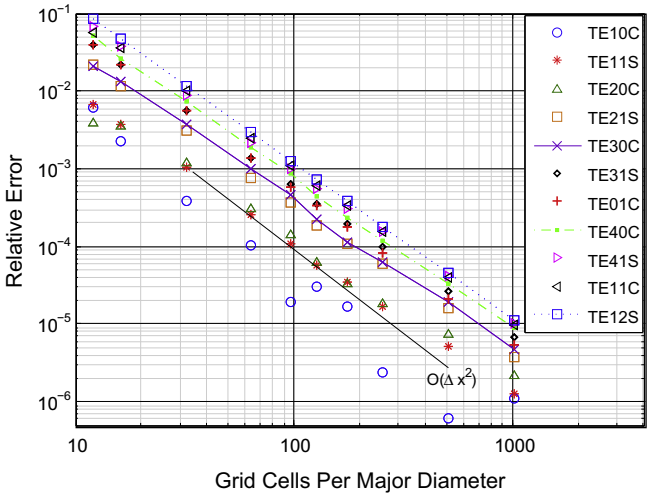

Fig. 10. The relative error in eigenvalue (frequency squared) versus grid resolution, for the lowest 11 TE modes of a 2D elliptical cavity, using $f_{DM} = 0.2$.

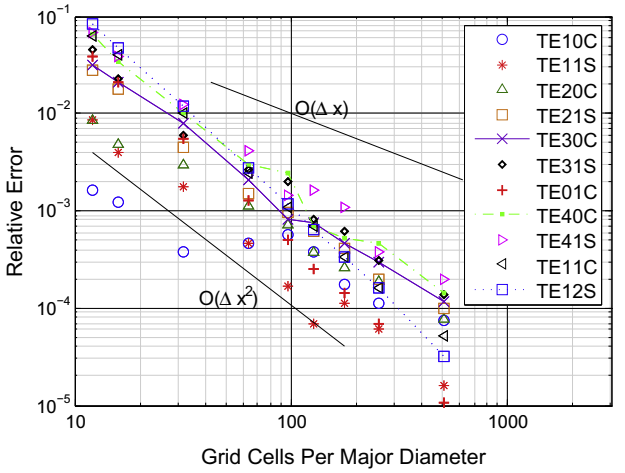

Fig. 11. The relative error in eigenvalue (frequency squared) versus grid resolution for the lowest 11 TE modes of a 2D elliptical cavity, using $f_{DM} = 0.5$.

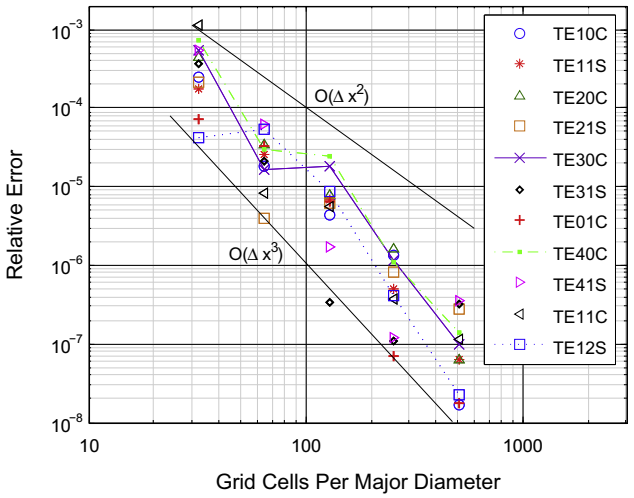

Fig. 12. The relative error in eigenvalue (frequency squared) versus grid resolution when all cut faces are included, after Richardson extrapolation assuming second-order error from simulations with the number of grid cells per major diameter, $N = 16$ and 32 up to simulations with $N = 256$ and 512 (Compare to Fig. 9).

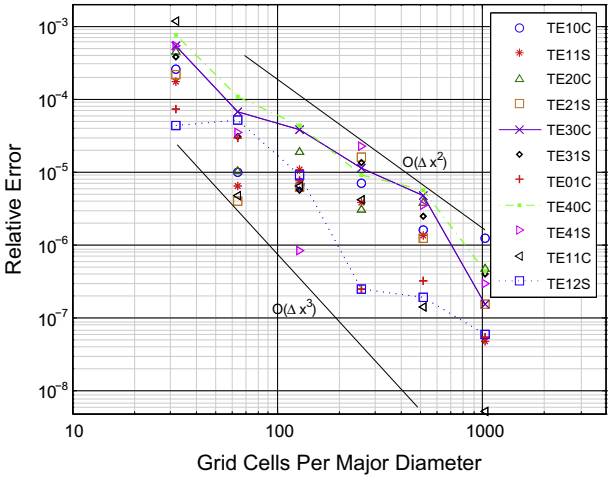

Fig. 13. The relative error in eigenvalue (frequency squared) versus grid resolution, using $f_{DM} = 0.2$, after Richardson extrapolation assuming second-order error, using simulations from $N = 16$ and 32 up to simulations with $N = 512$ and $N = 1024$. At coarse resolutions, the error seems roughly an order below that in Fig. 10, but at fine resolutions the error is just below second order.

(Figs. 13 and 14), because neglecting cut faces introduces an error that changes discontinuously with cell-size (as cell size changes, there is point where a partial cell is suddenly omitted, or suddenly included). In other words, when some cut faces are neglected, $f(\Delta x)$ is not a smooth function, so it cannot be usefully extrapolated.

8. Boundary modes

As discussed in Section A, the stable time step $\Delta t_{\max}$ is reduced from the value Δt_{CFL} for the infinite system. This is caused by the fact that the Dey–Mittra update introduces modes that have frequencies higher than those that can propagate throughout the interior of the cavity. Hence, those higher-frequency modes are trapped in the vicinity of the boundary. There are methods referred to collectively as area-borrowing methods [8,9] that eliminate the reduction of time step for stability. It follows that such methods also eliminate boundary modes; if non-propagating, higher-frequency modes were present, the reduction of the time step for stability would remain.

Fig. 15 shows the magnetic field from a simulation of a TE32 mode in a cylindrical cavity. Since it is a transverse electric mode the magnetic field is perpendicular to the plane of the plot. The solid and dashed lines are contours representing the analytical shape of the mode. The bright and dark spots at the edge of the cylinder are trapped boundary modes. In Fig. 16 we

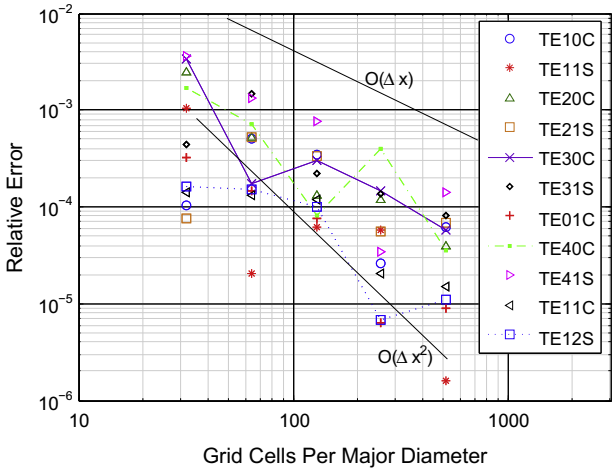

Fig. 14. The relative error in eigenvalue (frequency squared) versus grid resolution, using $f_{DM} = 0.5$, after Richardson extrapolation assuming second-order error.

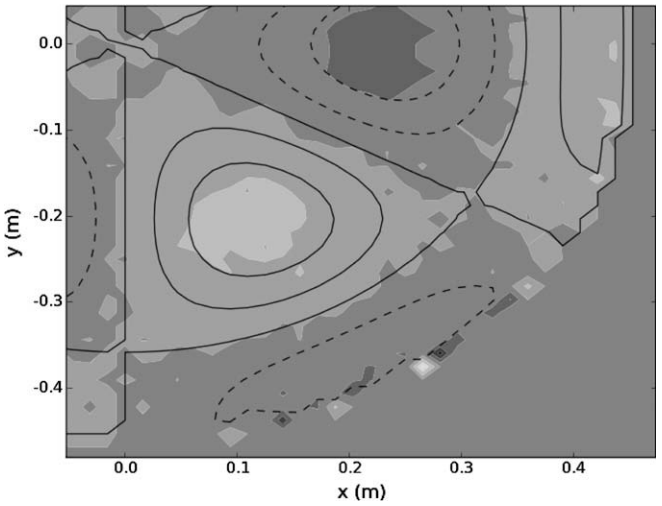

Fig. 15. Contour plot of the magnetic field for a simulation of a TE32 mode of cylindrical cavity. The bright peaks and dark troughs are trapped modes at the cylindrical boundary.

plot the magnetic field from the simulation as a function of the time. There are two dominant frequencies in the signal. The lower frequency of 0.86 GHz corresponds to the TE32 mode. The higher-frequency of 11.8 GHz is the frequency of the trapped mode. The cut off frequency for the computational grid in this simulation is 4.3 GHz, which is considerably lower than the frequency of the trapped mode.

For computations of pure electromagnetics, such boundary modes can typically be avoided. For example, the method, discussed in Section 5, of adiabaticallly ringing up the cavity does not excite these higher-frequency modes. Perturbations that are far from the boundaries also would not excite these modes. However, in particle simulations where particles can interact with the walls (reflection, absorption, emission) the excitation of these modes could be problematic. Such modes would be readily excited by the shot noise accompanying particle–wall interactions. Thus, the use of methods like those of Ref. [8,9] will be important for simulations involving particles.

9. Challenges applying Dey–Mittra methods to EM–PIC simulations

In the present work we have focused on testing the Dey–Mittra cut cell algorithm using the FDTD electromagnetic solver in VORPAL. VORPAL is also a Particle-In-Cell (PIC) code [10,11] used to represent plasmas. Many of the application areas where we plan to apply the Dey–Mittra method involve the presence of a plasma. Since these will involve the use of the

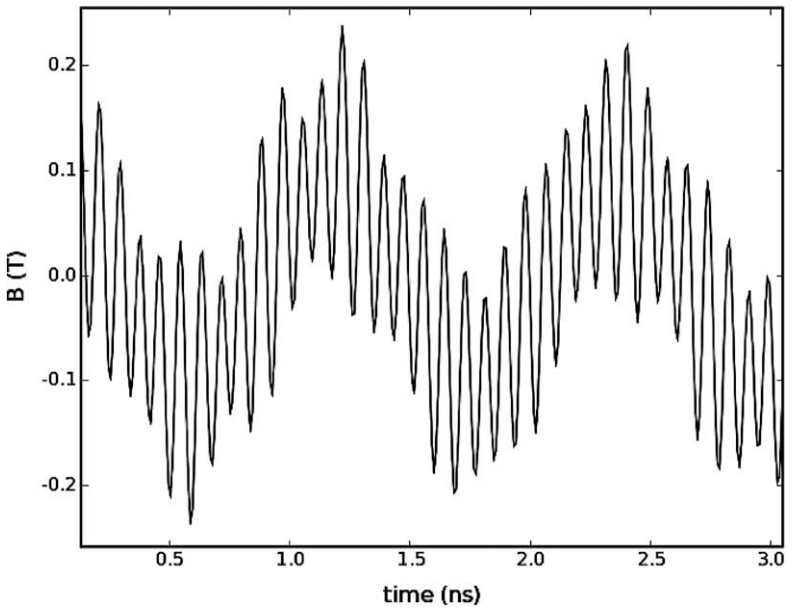

Fig. 16. Time series for the magnetic field for one of the trapped modes in the cylindrical TE32 mode simulation.

PIC model in VORPAL we discuss some of the issues that come up with cut cell electromagnetics when used in conjunction with PIC algorithms.

PIC methods use a collection of macro-particles to represent charged particles with an arbitrary velocity distribution. Rather than represent every particle in the distribution, each macro-particle represents a large collection of particles with similar velocities. Since the macro-particles represent multiple particles, their locations are the average location of the particles they represent. The associated currents produced by the macro-particles are assigned to the computational grid used by the Yee FDTD method by using area weighting to compute the amount crossing the dual-grid faces [1]. (The cells of the dual-grid are offset from the regular cells by one-half cell in each direction; the electric fields are located at the faces of the dual-grid cells.) Allowing particles to leave the simulation domain at the edge of grid does not present any problems, since the currents produced by the particles as they leave the domain ensure all the macro-particles charge has been removed from the simulation domain.

The situation is more challenging when a cut-cell boundary is involved. A macro-particle that crosses a cut-cell boundary has left the simulated region and, so, should be removed. However, at the point of crossing a cut-cell boundary, the charge of the particle will not necessarily have crossed all interior dual-grid interior surfaces. Thus, immediate removal can lead to ghost charge (non-zero divergence of the electric field). This can accumulate to an unphysical degree. Several strategies to eliminate this are possible, including immediate movement of the particle to an exterior regular grid node.

10. Summary and conclusions

The Dey–Mittra conformal boundary algorithm has given the VORPAL plasma simulation framework the ability to model electromagnetics in complex geometries that are usually very challenging for a FDTD code running with a Cartesian mesh. We have demonstrated that for global quantities such as the mode frequency of a cavity, the Dey–Mittra gives second order convergence with a weak first order tail that only impacts the results when the grid resolution becomes fine enough that small cut cells have to be replaced with stair-step boundaries to maintain stability. Since our implementation of the Dey–Mittra algorithm includes a detailed stability analysis, we introduce a parameter, f_{DM} which allows us to control the fraction of the full CFL time that will be stable for the algorithm. This parameter allows the user to balance the accuracy of the method with the computational resources needed to run a simulation.

At the end of Section 5 we have applied VORPAL's Dey–Mittra algorithm to the TESLA cavity design to be used in the International Linear Collider (ILC). We compare the mode frequencies found by VORPAL with experimental measurements of the cavity and find we can determine the number and frequencies of the cavity modes in the frequency band of interest.

The Dey–Mittra algorithm does have its drawbacks. The time step must be reduced relative to the Courant limit to maintain stability. Considering the gains made in the accuracy of the simulation and that the time step reduce is usually small (50%) this is minor concern. The existence of trapped modes could conceivably cause problems for applications where the physics at the boundary is being studied but the generation of these modes has not been seen much in practice. Area-borrowing methods [8,9] achieve a similar level of accuracy without the various disadvantages of the Dey–Mittra method, but they are more complex and difficult to implement.

Acknowledgments

This work was supported by the US Department of Energy Grant DE-FG02-04ER41317 and Air Force Office of Scientific Research Grants FA9550-04-C-0041 and FA9451-06-M-0095.

We also acknowledge assistance from the VORPAL team: T. Austin, G.I. Bell, D.L. Bruhwiler, R.S. Busby, J. Carlsson, M. Carey, B.M. Cowan, D.A. Dimitrov, A. Hakim, J. Loverich, S. Mahalingam, P. Messmer, P. Mullowney, K. Paul, C. Roark, S.W. Sides, N.D. Sizemore, R. Trines, S.A. Veitzer, D.J. Wade-Stein, W.-L. Wang, M. Wrobel, N. Xiang, and W. Ye, C.D. Zhou.

Appendix A. Implementation of conformal boundaries within VORPAL

VORPAL was designed from the beginning to support simulations of different dimensions with the same code base [12]. This means the finite difference update of the fields on the computational grid is not done with a simple set of nested loops as is the case in many other codes. A stencil that represents the finite difference operator is 'walked' over the grid by a class structure that understands the layout of the grid and how it varies with different dimensions. The actual field data is stored in a one dimension array and the walker classes know how move the stencil from point to point including skipping certain cells that are not updated, such as the guard cells that exists for parallel messaging.

Since this structure already supported the skipping of certain cells it was not too difficult to introduce the simulation of complex structures. The geometry information is stored in a separate class whose interface includes a method that determines if a point in space is inside or outside of the boundary. In the current implementation of VORPAL this information is determined by a function that defines the boundary. When the electromagnetic update is set up, each cell is checked to see if the cell is completely inside or outside of the boundary or if it the boundary cuts though the cell (making it a so called cut-cell).

Since VORPAL uses the Yee [2] staggered mesh the electromagnetic update is effectively 2D for both 2D and 3D simulations since only the curl equations are used and each of the difference equations involved is limited to field values that lie in plane. In particular the Faradary update only involves field components on the faces of the cells.

For more primitive stair-step boundaries if a face is cut by the boundary then it is kept if more than 50% of the face's area is interior. A similar class is then used to walk the stencils across the grid, knowing what cells to skip based on the cells that where identified during the set up. The Dey–Mittra method adds another level of complication since the stencils for the boundary faces depend on how the boundary is cut though the face. We created a new class that represents the Dey–Mittra stencil which is flexible enough to account for the differences between different cut faces. Two passes are now needed over grid each with different walkers. The first pass updates the interior cells and skips the boundary and exterior cells. The second pass updates the boundary cells, where each face has a unique stencil, and skips the interior and exterior cells.

The results of this stability analysis was incorporated into the Dey–Mittra implementation in VORPAL. The user specifies the f_{DM} parameter, determining how small his time step must be to maintain stability. During the set up of the updates for the boundary cells in the simulation each of the cut cells is matched with the associated stability condition and if the condition is not meet the cut cell is replaced with a stair-step boundary.

Following VORPAL's object oriented design, the relation of the conformal boundary surface and the grid is stored in a separate object. Currently the boundary surface for a VORPAL simulation is described by a function, $f(x,y,z) = 0$. We use the convention that when the function is negative that describes the interior of a conductor so no field update is done there and when the function is positive that is a vacuum region where an electromagnetic update is needed. The boundary object's interface provides a method to determine if a spatial point is inside or outside the vacuum region and data structures which list the index of all the grid cells that are cut by the boundary and the fractional lengths and area of all the cut edges and faces of the cut cells.

At start up when the data structures for the cut cell are determined the various stability conditions discussed above are checked against the f_{DM} parameter given in the input file. If a cell face is found to be unstable for that time step fraction then that cell face is not marked as a cut-cell face but is marked as either fully interior or fully exterior depending on how it is cut. This results in a stair-step boundary for that cell face.

The electromagnetic fields use the information from the grid boundary object to set up three different sets of stencils to be used in the finite difference update of the fields. One type of stencil corresponds to the Ampere update of the electric field. This stencil does not use the cut-cell information and is applied to all the interior faces as well as all the cut-cell faces. There are two types of stencils that correspond to the Faradary update of the magnetic field. The first is the standard stencil for the Yee method which is applied to the interior cells. The second represents the Faraday integral around the cut edges of a cut-cell and is only applied to the cut-cell faces.

References

[1] J. Villaseñor, O. Buneman, Rigorous charge conservation for local electromagnetic-field solvers, Comput. Phys. Commun. 69 (1992) 306.
[2] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas Propagat. 14 (1966) 302.
[3] S. Dey, R. Mittra, A locally conformal finite-difference time-domain (FDTD) algorithm modeling modeling three-dimensional perfectly conducting objects, IEEE Microwave Guided Wave Lett. 7 (1997) 273.

[4] Y. Shapira, Solving PDEs in C++, numerical methods in a unified object-oriented approach, SIAM, 2006.
[5] H. Padamsee, J. Knobloch, T. Hays, RF superconductivity for accelerators, Wiley-VCH, 2008.
[6] B. Aune, R. Bandelmann, D. Bloess, B. Bonin, A. Bosotti, M. Champion, C. Crawford, G. Deppe, B. Dwersteg, D.A. Edwards, H.T. Edwards, M. Ferrario, M. Fouaidy, P.-D. Gall, A. Gamp, A. Gössel, J. Graber, D. Hubert, M. Hüning, M. Juillard, T. Junquera, H. Kaiser, G. Kreps, M. Kuchnir, R. Lange, M. Leenen, M. Liepe, Superconducting TESLA cavities, Phys. Rev. ST Accel. Beams 3 (9) (2000) 092001.
[7] The TESLA Technology Corporation, Cavity Database, <http://tesla-new.desy.de/cavity_database/homs__pickups/index_eng.html> (last accessed October 2007).
[8] I.A. Zagorodnov, R. Schuhmann, T. Weiland, A uniformly stable conformal FDTD-method in Cartesian grids, Int. J. Numer. Model. 16 (2003) 127.
[9] R.J. Barker, N.C. Luhmann, J.H. Booske, G.S. Nusinovich, Modern Microwave and Millimeter-Wave Power Electronics, IEEE, 2005.
[10] C.K. Birdsall, A.B. Langdon, Plasma Physics Via Computer Simulation, Adam Hilger, 1991.
[11] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, Adam Hilger, 1988.
[12] C. Nieter, J.R. Cary, VORPAL: a versatile plasma simulation code, J. Comput. Phys. 196 (2004) 448.