JOURNAL OF COMPUTATIONAL PHYSICS 68, 48–55 (1987)

A Method for Incorporating Gauss' Law into Electromagnetic PIC Codes

BARRY MARDER

Simulation Theory Division 1231, Sandia National Laboratories, P. O. Box 5800, Albuquerque, New Mexico 87185

An algorithm is presented which solves a modified set of Maxwell's equations, none of them elliptic, in an electromagnetic PIC type code. The algorithm prevents large violations of Gauss' law by diffusing away numerical errors arising from the particle-to-grid apportionment method. There are no restrictions on the manner in which charge and current densities are determined from the particle distribution. In particular, point particles and simple linear interpolation are used. © 1987 Academic Press, Inc.

Sandia National Laboratories makes extensive use of two-dimensional electromagnetic particle-in-cell (PIC) codes for plasma simulation and is currently engaged in a project to develop a general purpose three-dimensional code. The algorithm described here is under consideration both for use in the 3-D code and for implementation in existing 2-D codes.

Faraday's and Ampere's laws (in dimensionless variables),

$$\partial \mathbf{B}/\partial t = -\text{curl } \mathbf{E} \tag{1}$$

$$\partial \mathbf{E}/\partial t = \text{curl } \mathbf{B} - \mathbf{J}, \tag{2}$$

are mathematically consistent with Gauss' law,

$$\text{div } \mathbf{E} = \rho, \tag{3}$$

provided

$$\text{div } \mathbf{J} + \partial \rho/\partial t = 0. \tag{4}$$

To solve these equations in problems in which there is plasma flow, particle-in-cell codes are often used. In these codes, the plasma is represented by a large number of numerical "particles" which carry mass and charge and move through an underlying grid. The grid stores ensemble averages such as charge and current densities, as well as electric and magnetic fields obtained from differential equations. There is a constant interplay between the grid and particles as the grid quantities are used to define the forces which move the particles, and the particles send position and velocity information back to the grid.

0021-9991/87 $3.00

Perhaps the simplest method for computing the charge and current densities in a PIC code is to treat the particles as idealized point sources of charge and use linear interpolation (also called area-weighting) to assign these quantities to the corners of the cell in which the particle is found. This technique may be thought of alternatively as using polygonal particles (for example, cubes) in which the volume overlapping a cell is apportioned to that cell. The assignment of charge by this method is continuous as the particle moves through the grid. When current and density are determined by this method, however, (4) will, in general, be only approximately satisfied. If the electric and magnetic fields are then determined from (1) and (2), (3) will not be solved exactly, As the code runs, discrepancies between the charge and the divergence of the electric field could develop.

In these calculations, in which the electric fields are determined from their time derivatives while charge and current densities are obtained from the particle distribution, Gauss' law resembles an energy integral in that it is derivable from the equaions used to solve the problem but is not needed for the calculation. Energy integrals are used as a check of a code's accuracy. If the code is consistent with the equations and is well written, the energy error should be small. Gauss' law should also be nearly satisfied and could serve as another accuracy check. However, because it enjoys the status of one of Maxwell's equations, and indeed becomes the governing equation in electrostatic problems, it is felt that more attention should be given to it.

Three basic approaches [1] have been taken to include Gauss' law in electromagnetic codes. In the first method, either the vectors are decomposed into curl-free and divergence-free form or the vector potentials are used. Poisson's equation is solved at every time step for the electrostatic component of the field. There are no restrictions on the particle-to-grid apportionment technique. While these methods accurately reflect the physics, solving Poisson's equation at every time step can be a very time-consuming operation in a general purpose three-dimensional code.

A second approach is to solve (1) and (2) as they stand but apply a Poisson solver correction to the electric field after a certain number of time steps. If the "fixes" are applied too frequently, however, these codes will still be very time-consuming. If they are applied too infrequently, abrupt discontinuities may be introduced. When this "fix" is applied, of course, the electric field is no longer determined solely by Eq. (2). One has sacrificed Ampere's law for Gauss'. In any case, it is aesthetically more pleasing to perform the same operations every time step.

The third method for satisfying (3) is to compute the charge and current densities in such a manner that (4) is exactly satisfied. The MAGIC [2] code uses this approach. This will, in general, require abandoning the simple linearly interpolated point particle method for a noisier, more complicated one. In these algorithms, a particle is considered to occupy a volume the size of the gird cell which contains it. When a particle intersects a grid vertex, charge and current densities are assigned there. Thus, only one grid point, that nearest the particle center, is affected by it. The on-off nature of this nearest-grid-point apportionment technique introduces numerical noise into the computation. To facilitate the exact solution of Eq. (4),

particles in MAGIC move parallel to the underlying grid. Thus, there are two ways a particle can traverse a diagonal path in a two-dimensional calculation and six available paths in three dimensions.

To avoid the type of problems inherent in these algorithms, a simple new method for solving Maxwell's equations in electromagnetic PIC codes has been developed which

(a) limits the buildup of error in Gauss' law,

(b) places no restriction on how the charge and current densities are obtained from the particles,

(c) does not require solving an elliptic equation,

(d) performs the same operations at each time step, and

(e) is readily extendable to three dimensions.

Define

$$F(x, t) = \text{div } \mathbf{E} - \rho. \tag{5}$$

Gauss' law becomes simply $F = 0$. For the numerical solution, (2) is altered by adding a multiple of the gradient of F to the right-hand side:

$$\partial \mathbf{E}/\partial t = \text{curl } \mathbf{B} - \mathbf{J} + d \text{ grad } F. \tag{6}$$

This added term will be referred to as a "pseudo-current." Notice that this is mathematically consistent with Maxwell's equations; one equation has simply been added to another. d is a numerical parameter chosen small enough not to affect adversely the stability but large enough to perform the desired function. The computation of grad F in the right-hand side of (6) requires only minimal additional computer time as most of the time in PIC codes is used in the particle loops. It can be easily shown that F satisfies the inhomogeneous diffusion equation

$$\partial F/\partial t - d\nabla^2 F = -(\partial \rho/\partial t + \text{div } \mathbf{J}). \tag{7}$$

The source term for F on the right-hand side of (7) reflects the extent to which the charge and current densities as defined on the grid satisfy the charge conservation equation, (4). It should be pointed out that exact charge conservation is automatic in PIC codes as the particles themselves carry the charge. A non-zero forcing term in (7) does not reflect charge creation or destruction, nor any other physical phenomenon, but is simply an artifact of the particular finite differencing and particle-to-grid apportionment technique. By virtue of (7), when F vanishes on the boundary, the code "diffuses" the error away. The parameter d, which governs the speed at which F diffuses, should be chosen so that the rate at which F is generated by the code is roughly balanced by the rate at which it dissipates. This will insure that, although F does not vanish identically, it does not build up as the code runs, but remains acceptably small. The algorithm is, in a sense, self-healing.

The stability restriction introduced by this addition is the well-known heat equation constraint [3]

$$2d\,\Delta t/\Delta x^2 < 1, \tag{8}$$

where Δx and Δt are the numerical space and time steps. In practice, d will usually be considerably below this limit. Other than this obvious limitation, no adverse effects on stability have been observed from this technique. Adding too much diffusion, however, can suppress the very physics the code is attempting to model. Adding too little, or none at all, allows the density obtained from the particles to differ from that obtained from the divergence of E to what may be an unacceptable degree. This is most noticeable in a vacuum cell where, although div E should vanish, it could remain non-zero because of the earlier presence of plasma.

For any domain in which both density and current vanish on the boudary, and for which no particle creation occurs inside, the numerical integral of the forcing term in (7) vanishes. There is then no net generation of F inside the region. Local numerical errors of one sign are compensated for by errors of the opposite sign elsewhere in the domain. The pseudo-current can then be thought of as a diffusive flow of electric field which tends to elimate these errors.

Because a gradient has been added to (2), the wave equation for B, obtained by combining (1) and the curl of (2),

$$\partial^2\mathbf{B}/\partial t^2 = \nabla^2\mathbf{B} + \text{curl}\,\mathbf{J} \tag{9}$$

remains unchanged both mathematically and numerically (assuming the curl of a gradient vanishes in the finite difference scheme). The magnetic component of electromagnetic waves propagates exactly as before. Only the electric field component is affected. In the pure vacuum case, of course, F remains identically zero.

The time-difference analog of (6) is

$$\mathbf{E}^{k+1/2} - \mathbf{E}^{k-1/2} = \Delta t[\text{curl}\,\mathbf{B}^k - \mathbf{J}^k + d\,\text{grad}\,F^{k-1/2}], \tag{10}$$

where k indicates discrete time steps. This equation correctly centers the **B** and **J** terms but not the F term, resulting in only first-order accuracy for this term. If it were deemed necessary, the explicit numerical solution of (6) can be made second-order accurate in time either by using a particular value of d or another differencing technique. It does not seem necessary to do this for the following reason. Rather than having added grad $F(x, t)$ to (2), suppose we had added grad $F(x, t-\Delta t/2)$. We would still have appended one of Maxwell's equations to a modified form of another, but the difference equation, (10), would now be second-order accurate. Equation (7) would, of course, be slightly altered. The point of this semantic quibble is that, despite the appearance of lost accuracy, the formal second-order accuracy of the original algorithm has, in fact, not been degraded.

This algorithm has been tested by simulating numerically several microwave generating devices, including backward wave oscillators, cross-field amplifiers, and

normal and inverted magnetrons [4]. The code appears to perform quite well, accurately reproducing the experimentally observed behavior of these devices. For illustrative purposes the simulation of the Stanford magnetron [5] will be discussed here. The magnetron is a cylindrical device with a center cathode and coaxial anode. The anode is grooved to lower the phase velocity of azimuthally traveling electromagnetic waves. An applied axial magnetic field insulates the diode. The voltage causes electrons to be emitted from the cathode which then undergo an $\mathbf{E} \times \mathbf{B}$ drift around the device. When the drift velocity of these electrons is equal to the phase velocity of the electromagnetic wave, the electrons can feed energy into the wave, generating microwaves. Three simulations of the same device will be presented to illustrate the algorithm. The first has no pseudo-current, the second has an amount deemed sufficient to keep Gauss' law reasonably satisfied, and the third has an excessive amount of pseudo-current. The grid in these examples consists of 34 nodes in radius and 43 nodes in theta. The time step was taken to be 85% of its Courant stability limit. Particles are emitted from the cathode and about 600 are present at saturation. The cathode radius is 2.54 cm, the anode radius is 5.08 cm, the depth of the voids is 1.74 cm, and the void angle is 14.5 degrees. The applied voltage is 500 kV and the magnetic field is 0.155 Tesla. There are 12 vanes in the full device; sixfold periodicity is assumed. With these parameters, a run takes about an hour of CPU time on a VAX 8600.

The three values of d in the code are 0, 0.001, and 0.01. To express these in physical units, (6) is written in dimensional form:

$$\varepsilon_0 \, \partial \mathbf{E}/\partial t = \operatorname{curl} \mathbf{B}/\mu_0 - \mathbf{J} + d \operatorname{grad}(\varepsilon_0 \operatorname{div} \mathbf{E} - \rho). \tag{11}$$

The pseudo-current parameter, d, has dimensions of length squared divided by time. The actual value of d is obtained by multipling the code values by the length scale (cathode radius = 2.54 cm) times the velocity scale (speed of light).

Figure 1 shows the quasi-equilibrium state of the magnetron at about 1 ns.

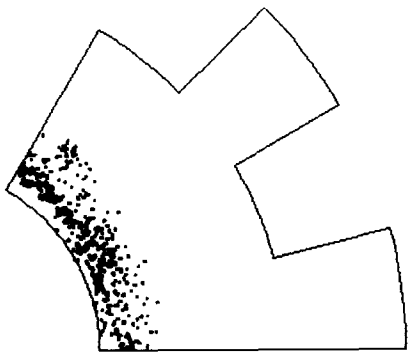

FIG. 1. Electron distribution in the Stanford magnetron simulation at about 1 ns. Inner circle is the cathode.

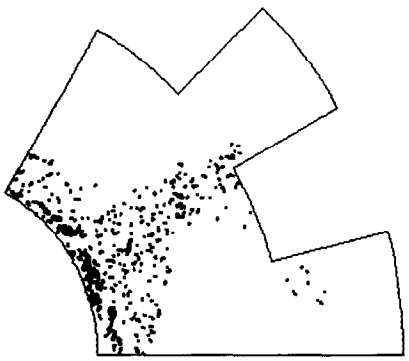

FIG. 2. Flow at about 15 ns.

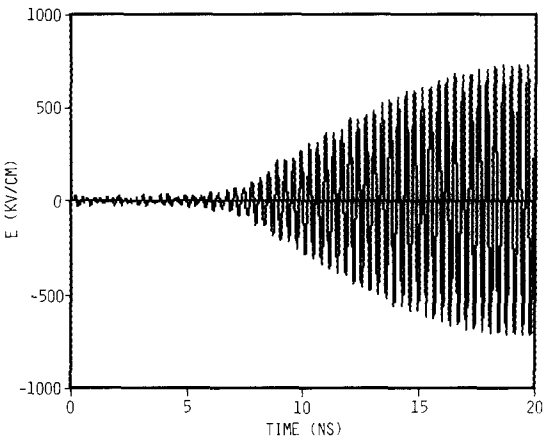

FIG. 3. RF field in void between vanes for $d = 0$.

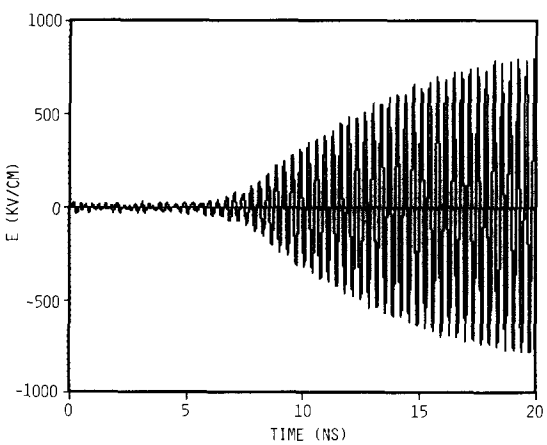

FIG. 4. RF field in void between vanes for $d = 0.001$

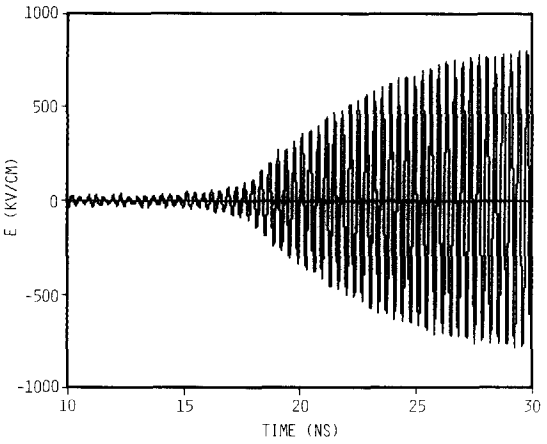

FIG. 5. RF field in void between vanes for $d = 0.01$. Note shifted time axis.

Figure 2 shows the saturated state at about 15 ns in which electrons form "spokes" which circle the device (counterclockwise in the figure). The oscillation frequency is consistent with the experiment, and the equilibrium and spoke development correspond to those seen in magnetron simulations using other codes [6, 7].

Figures 3, 4, and 5 are plots of the RF field in the void between the vanes for the three cases. Notice that since the pseudo-current has a smoothing effect on the calculation, the onset of instability takes longer in the third case. Once it starts, however, its growth rate and frequency are the same as the previous runs. Figure 6 shows the maximum F in the domain divided by the maximum density in the domain for the three cases. It does not take much pseudo-current to dramatically

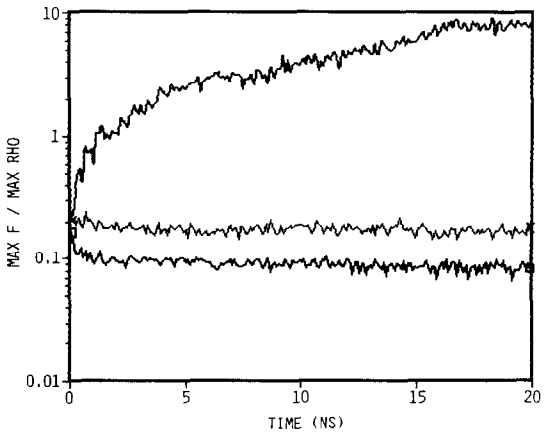

FIG. 6. Maximum violation of Gauss' law divided by the maximum density for $d = 0$, 0.001, and 0.01.

reduce this error. The observation that the first two simulations look almost identical reflects the fact that, for the length of time this problem ran, the error in Gauss' law did not strongly affect the solution. Although it is not obvious on the log plot, this error grows linearly in the first run and were the simulation to continue much longer its validity would have to be questioned. The error in the second run remains small and essentially constant throughout the run. This seems the ideal choice for d. With this choice for d, the pseudo-current terms are about an order of magnitude below the current density terms in the equations.

The algorithm presented here solves a modified set of Maxwell's equations, none of which are elliptic. In doing so, it bounds Gauss' law by diffusing away errors arising from particle-to-grid apportionment techniques. If sufficient particles are present and the apportionment scheme is consistent with the underlying physics, these errors should not be large to begin with, but without the pseudo-current, they could build up in time. This technique places no restriction on the method in which charge and current densities are determined from the particle distribution.

REFERENCES

1. C. K. BIRDSALL AND G. B. LANGDON, *Plasma Physics via Computer Simulation* (McGraw–Hill, New York, 1985), p. 359.
2. B. GOPLIN, R. E. CLARK, J. MCDONALD, AND W. MICHAEL BOLLEN, "Users Manual for MAGIC," Mission Research Corporation Report No. MRC/WDC-R-068, Alexandria, VA, September 1983.
3. R. D. RICHTMYER AND K. W. MORTON, *Difference Methods for Initial Value Problems* (Wiley, New York, 1967), p. 12.
4. J. C. SLATER, *Microwave Electronics* (Dover, New York, 1950).
5. W. P. BALLARD, S. A. SELF, AND F. W. CRAWFORD, *J. Appl. Phys.* **53**, 7580 (1985).
6. A. PALEVSKY, G. BEKEFI, AND A. DROBOT, *J. Appl. Phys.* **52**, 4938 (1981).
7. J. W. POUKEY, Sandia Report SAND83-1864, Sandia National Laboratories, P. O. Box 5800, Albuquerque, NM 87185, April 1984.