Virtual Prototyping of Radio Frequency Weapons

Keith L. Cartwright,* Andrew D. Greenwood, Peter J. Mardahl, Tony Murphy, and Michael D. Haworth

Center for Plasma Theory and Computation, Directed Energy Directorate, Air Force Research Laboratory, Kirtland Air Force Base, New Mexico 87117

This paper explains how radio frequency (RF) systems, from pulsed power through to antennas, can be virtually prototyped with the ICEPIC (Improved Concurrent Electromagnetic Particle-in-Cell) code. ICEPIC simulates from first principles (Maxwell's equations and Lorenz's force law) the electrodynamics and charged-particle dynamics of the RF-producing part of the system. We discuss features such as multiprocessing on distributed memory machines, dynamic load balancing, and the mitigation of numerical Cerenkov radiation in the field solve. Simulations focus on gigawatt-class sources; shown as an example will be the magnetically insulated line oscillator (MILO). Such simulations require enormous computational resources. Our simulations successfully expose undesirable features of these sources and have helped us to suggest improvements.

Nomenclature

B	magnetic field vector
c	speed of light
E	electric field vector
$H_b(\omega)$	filter function applied to B
$H_e(\omega)$	filter function applied to E
J	current vector
k	wavenumber
m	mass
N_{cell}	number of cells
$N_{particle}$	number of particles
q	electrostatic charge
t	time
v	velocity vector
W_{cell}	amount of computational work to update one cell
$W_{particle}$	amount of computational work to update one particle
$W_{subdomain}$	amount of computation work in a simulation partition
x	first Cartesian spatial coordinate
y	second Cartesian spatial coordinate

z	third Cartesian spatial coordinate
$\alpha_1, \alpha_2, \alpha_3$	coefficients used for filtering
γ	relativistic factor
Δ_t	discrete timestep
$\Delta_{t,max}$	maximum timestep for algorithm stability
Δ_x	grid spacing in the x direction
Δ_y	grid spacing in the y direction
Δ_z	grid spacing in the z direction
∂	partial derivative
θ	filtering parameter
ρ	charge density
ω	angular frequency
∇	gradient operator
$\times$	cross product of vectors, or multiplication

1. Introduction

Directed energy devices project energy in a desired direction at the speed of light to perform a useful task. The most ubiquitous and well-known example of this is the laser, which operates at optical frequencies. However, in principle, electromagnetic energy of any frequency can be emitted in a focused beam. In addition to lasers, the Directed Energy Directorate of the Air Force Research Laboratory is investigating the production of focused electromagnetic radiation in the microwave and millimeter-wavelength portions of the electromagnetic spectrum. There are several applications for high-power microwave (HPM) energy, including radio frequency (RF) weapons for nonlethal attack on electronics; drivers for medical, industrial, and high-energy accelerators; remote imaging; materials processing for enhanced sintering; power beaming for novel power grid architectures; and space propulsion such as microwave beam sailing.

The goal of this paper is to give researchers a predictive capability in the design of HPM devices. The focus of this effort is on development and application of new particle-in-cell[4] (PIC) plasma physics simulation software that runs efficiently on parallel high-performance computing (HPC) platforms. We call this software ICEPIC (Improved Concurrent Electromagnetic Particle-in-Cell).[22]

Devices that generate and deliver electromagnetic waves in the microwave frequency band (generally thought to be 0.1–100 GHz) hold great promise for a number of military applications, including the disruption or disabling of enemy electronics,[3] and for commercial applications, including beamed energy for space propulsion.[2] The focus of this effort is RF weapons. As weapon systems become increasingly dependent on electronic components, they also become more susceptible to microwave radiation, in turn increasing the importance of HPM sources. Sources currently under development at locations such as the Directed Energy Directorate of the Air Force Research Laboratory are capable of producing gigawatt power levels for a few hundred nanoseconds. Future requirements demand that power levels, pulse lengths, and/or rep-rates be increased and that the size of the devices that presently occupy large laboratory bays be decreased. Thus, significant development remains to be done for HPM devices.

A typical HPM device may be considered to have three main components: a pulsed power source, a microwave source (containing a diode and resonant structure), and a radiating structure (or "antenna"). The pulsed power source is used to create a dc voltage that is applied

to a diode to generate an electron beam and to accelerate it to relativistic energy. The resonant structure, which may be a slow-wave structure or a series of cavities, converts kinetic and/or potential energy in the beam to electromagnetic energy in the form of microwaves. The radiating structure directs the microwaves toward a target and acts as the mechanical interface between the atmosphere and the microwave source, which typically operates under vacuum.

Historically, development of microwave sources is accomplished primarily through experimental trial and error, with some theoretical guidance. As sources mature and achieve successively higher powers, the models describing the governing phenomena become too complex for closed-form analytical solution. Consequently, modern source development relies heavily on numerical modeling techniques. In fact, some sources are conceived and developed entirely using the PIC numerical method.[1] Simulations also guide the development of existing sources to extend their range of operation and to help understand and overcome their limitations.

There are several possible benefits to a device design process that relies heavily on simulations. First, the quicker turnaround time in trying new designs and modifications results in cost savings. In addition, simulation facilitates deeper understanding of the principles that govern the operation of a device because components or physical processes may be isolated. Finally, every physical quantity that is computed may be diagnosed to within the resolution of the simulation; only limited information is available from experiments. These benefits suggest that it is necessary to continue improving numerical models and applying them to the development of microwave devices.

PIC techniques are appropriate for plasma phenomena such as those that are important in microwave sources. In the next section, we review briefly the physics, numerical methods, and performance on Department of Defense (DoD) HPC assets of the ICEPIC software. In this paper, we describe our success at applying ICEPIC to study the magnetically insulated line oscillator (MILO).[13]

2. ICEPIC Overview

ICEPIC is a three-dimensional Cartesian coordinate PIC code developed at the Air Force Research Laboratory to support the development of HPM sources. It solves Maxwell's equations and Lorentz's force law in the time domain. The goal for developing ICEPIC is to create a portable HPM simulation tool using modern parallel computing techniques. The software is written entirely in ANSI standard C and uses an intermediate message-passing library that links to any of several common libraries (including MPI and PVM). With the emergence of MPI as the industry standard, ICEPIC is now solely based on MPI. The code incorporates numerous advances in parallel computing. ICEPIC performs automatic partitioning using a single domain decomposition algorithm, asynchronous message passing, and dynamic load balancing. Typical ICEPIC applications employ several million cells and several hundred thousand particles. Processing time for a single simulation usually takes several tens of hours.

3. Governing Equations

ICEPIC computes the time advance of the magnetic field B according to Faraday's law and the electric field E according to Ampere–Maxwell's law. Both equations are solved in

Heaviside–Lorentz's form (bold notation indicates vectors):

$$\frac{\partial \boldsymbol{B}}{\partial t} = -c\nabla \times \boldsymbol{E}, \tag{1}$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = c\nabla \times \boldsymbol{B} - \boldsymbol{J}. \tag{2}$$

$\boldsymbol{J}$ represents the current density, and c is the speed of light. The particles used in ICEPIC are "macro-particles" that represent many charged particles (electrons and/or ions) with a position x and velocity vector $\boldsymbol{v} = \mathrm{d}x/\mathrm{d}t$. The relativistic form of Lorentz's force equation is used to determine a particle's velocity:

$$\boldsymbol{F} = m\frac{\mathrm{d}\gamma \boldsymbol{v}}{\mathrm{d}t} = q\left(\boldsymbol{E} + \frac{1}{c}\boldsymbol{v} \times \boldsymbol{B}\right). \tag{3}$$

Time is represented by t, and γ is the usual relativistic factor $(1 - v^2/c^2)^{-1/2}$. The particle's mass is represented by m, and q represents the particle's charge. It is the tradition in PIC to define the ratio q/m for the particles. The discrete forms of the dynamical equations (1) and (2) are designed to preserve the constraint equations $\nabla \cdot \boldsymbol{B} = 0$ and $\nabla \cdot \boldsymbol{E} = \rho$ as long as the initial data satisfy these constraints.

4. Numerical Methods

ICEPIC uses a Cartesian grid to difference the electric and magnetic field equations [Eqs. (1)–(2)]. The vector quantities $\boldsymbol{E}$, $\boldsymbol{B}$, and $\boldsymbol{J}$ are staggered in their grid location using the technique of Yee.[4,28] $\boldsymbol{E}$ and $\boldsymbol{J}$ are located on the edges of the primary grid, whereas $\boldsymbol{B}$ is located on the faces of the primary grid. An explicit leapfrog timestep technique is used to advance the electric and magnetic fields forward in time. The leapfrog method has the advantages of simplicity and second-order accuracy. The electric field advances on whole-integer timesteps, whereas the magnetic field and the current density advance on half-integer timesteps.

The three components of the momentum and position of each particle are updated via Eq. (3), using the Boris relativistic particle push algorithm[5] after advancing the field components. The particle equations for velocity and position are advanced also with a leapfrog technique. The velocity components are advanced on half-integer timesteps, and the particle positions are updated on the whole-integer timesteps. The current density weighting is based on an exact charge-conserving current weighting algorithm by Villasenor–Buneman,[27] enforcing $\nabla \cdot \boldsymbol{E} = \rho$. Once the particle's positions and velocities are updated and the new current density is updated on the grid, the solution process starts over again by solving the field equations.

5. ICEPIC Field Solver

The goal of device simulation is obviously to replicate the relevant device physics as closely as possible. This sometimes requires new physical mechanisms to be added to the code. Further, existing physical algorithms are examined with the intention of improving their performance. Thus, the Yee field solve algorithm is examined with the goal of improving device performance.

Electromagnetic radiation in free space propagates with the physical dispersion relation

$$\frac{\omega}{c} = k, \qquad (4)$$

where ω is the temporal angular frequency and k is the spatial wavenumber. Electromagnetic radiation simulated by the Yee algorithm propagates according to the dispersion relation

$$\frac{1}{(c\Delta_t)^2} \sin^2\left(\frac{\omega\Delta_t}{2}\right) = \left[\frac{1}{\Delta_x^2} \sin^2\left(\frac{k_x\Delta_x}{2}\right) + \frac{1}{\Delta_y^2} \sin^2\left(\frac{k_y\Delta_y}{2}\right) + \frac{1}{\Delta_z^2} \sin^2\left(\frac{k_z\Delta_z}{2}\right)\right], \quad (5)$$

where Δ_x, Δ_y, Δ_z, and Δ_t are the grid size in x, y, z, and the simulation timestep, respectively. For numerical stability,[4] the simulation time step must satisfy

$$\Delta_t \le \frac{1}{c}\left[\left(\frac{1}{\Delta_x}\right)^2 + \left(\frac{1}{\Delta_y}\right)^2 + \left(\frac{1}{\Delta_z}\right)^2\right]^{-1/2}. \qquad (6)$$

Note that the Yee algorithm numerical dispersion relation is anisotropic, that is, it is a function of the wave propagation direction. Nevertheless, the Yee algorithm numerical dispersion relation approximates the physical dispersion relation at long wavelengths (small wavenumber k), as shown in Fig. 1, where waves propagating along a grid axis, a two-dimensional (2D) grid diagonal, and a three-dimensional (3D) grid diagonal are shown and $\Delta_x = \Delta_y = \Delta_z = \Delta$ and $\Delta_t = v\Delta_{t,\max}$. It is observed from Fig. 1 that if Δ is sufficiently small relative to the shortest wavelength of physical interest in the simulation, the physics of interest are accurately represented. However, radiation at shorter wavelengths can exist
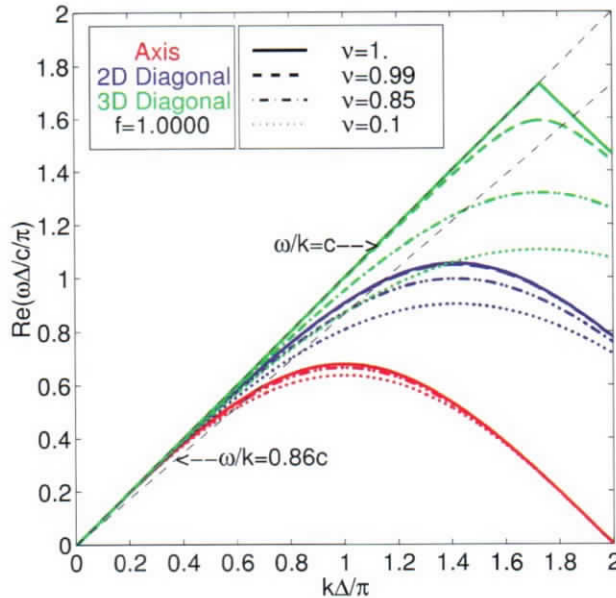

Fig. 1. Numerical dispersion relation of Yee algorithm showing three wave directions and four fractions of the maximum timestep.

on the simulation grid. The grid poorly resolves this short-wavelength radiation, and it propagates significantly slower than the physical speed of light c.

Because the poorly resolved, short wavelengths in the simulation are not of physical interest, it is tempting to assume that the nonphysically slow propagation speed is of little or no consequence. However, suppose that high-energy particles with velocities up to $0.86c$ are also of interest in the simulation. Figure 1 shows that the poorly resolved, short-wavelength radiation on the grid travels slower than these high-energy particles, leading to numerical (nonphysical) Cerenkov radiation. Because the interaction of the particles with the fields is nonlinear, the high-frequency numerical Cerenkov radiation couples to lower frequencies and corrupts the simulation at the frequencies of physical interest. A method of mitigating the nonphysical, numerical Cerenkov radiation is necessary for accurate simulation results.

The common method of mitigating numerical Cerenkov radiation is to use the time-biased iterative field solve introduced by Godfrey.[10] The algorithm replaces the values of B at timestep $n + 0.5$ that are used in Eq. (2) to calculate E at timestep $n + 1$ with $\alpha_1 B^{n+1.5} + \alpha_2 B^{n+0.5} + \alpha_3 B^{n-0.5}$, where $\alpha_1 + \alpha_2 + \alpha_3 = 1$ and $B^{n+0.5}$ denotes B at timestep $n + 0.5$. Typically, the algorithm is used with $\alpha_3 = 0$, and iteration is necessary to include the noncausal $B^{n+1.5}$ term. To understand the physical effect of this algorithm, it is interpreted as a digital filter applied to B in time. The physical effect of filters is in turn analyzed by deriving the numerical wave dispersion relation with a filter $H_e(\omega)$ applied to E and a filter $H_b(\omega)$ applied to B:

$$
\frac{1}{H_e(\omega)H_b(\omega)c^2\Delta_t^2} \sin^2\left(\frac{\omega\Delta_t}{2}\right) = \left[\frac{1}{\Delta_x^2} \sin^2\left(\frac{k_x\Delta_x}{2}\right)\right.
$$
$$
\left. + \frac{1}{\Delta_y^2} \sin^2\left(\frac{k_y\Delta_y}{2}\right) + \frac{1}{\Delta_z^2} \sin^2\left(\frac{k_z\Delta_z}{2}\right)\right]. \tag{7}
$$

Note that the dispersion relation sees no difference between a filter applied to E and one applied to B. The fact that the filter responses show up multiplying the wave propagation speed c indicates that the magnitude of the filter response alters the wave propagation speed while the phase of the response introduces wave damping or wave growth, depending on its sign. The ideal filter to limit numerical Cerenkov radiation causes wave damping at the poorly resolved, high frequencies on the simulation grid and allows the frequencies of physical interest to pass unaltered. Note that the ideal filter is not a traditional low-pass filter.

The response of the filter used by the time-biased field solve with $\alpha_1 = 0.125$, $\alpha_3 = 0$, is shown in Fig. 2. Note that the magnitude response of the filter is low pass, which further slows the propagation of the high-frequency waves that lead to numerical Cerenkov radiation. However, the phase response of the filter attenuates these waves, which effectively eliminates the Cerenkov radiation problem. The phase response of the filter is also linear at low frequency, indicating that physical frequencies of interest are also attenuated. It turns out that the use of this filter increases the maximum stable timestep for the simulation. However, the increased timestep size does not compensate for the additional time required by the iterations that are needed to include the noncausal B term. All of the features of the algorithm discussed above can be seen in the dispersion relation plots in Fig. 3.

The time-biased, iterative field solve, while commonly used for PIC simulation, has a number of undesirable features such as attenuation at the physical frequencies of interest
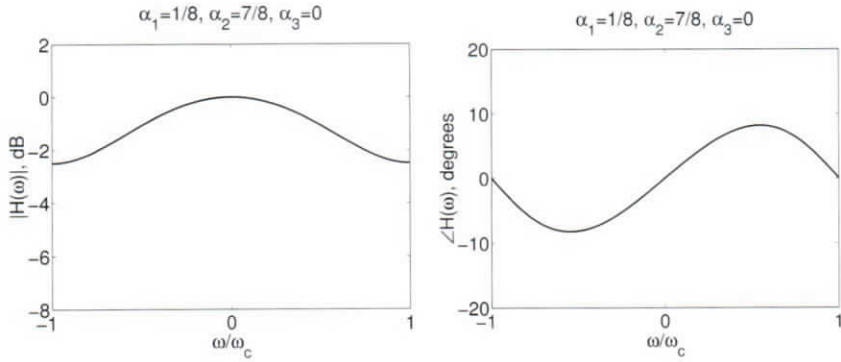
Fig. 2. Response of filter used by time-biased, iterative field solve. The Nyquist cutoff frequency is ω_c.
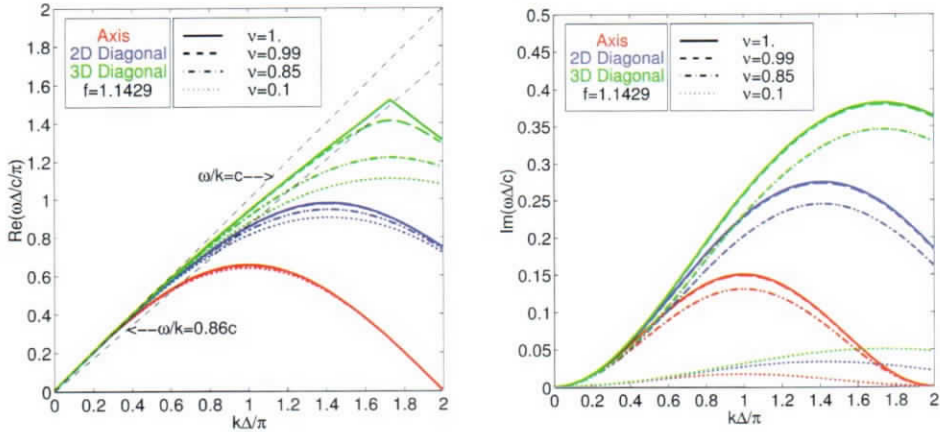

Fig. 3. Dispersion relation of the time-biased field solve algorithm with $\alpha_1 = 0.125$ and $\alpha_3 = 0$; the maximum stable timestep is increased by a factor of $f = 1.1429$ from the maximum stable timestep of the Yee algorithm.

and the extra time taken by the iterations. Using the filter interpretation, it is possible to analyze several algorithms and to compare their performances.

The filter interpretation is thus used to analyze the algorithm introduced by Friedman.[9,21] This method replaces the values of E at time step n used in Eq. (1) to compute B at time step $n + 0.5$ with

$$\left(1 + \frac{\theta}{2}\right) E^n - \theta \left(1 - \frac{\theta}{2}\right) E^{n-1} + \frac{1}{2}(1 - \theta)^2 \bar{E}^{n-2}, \tag{8}$$

where

$$\bar{E}^{n-2} = E^{n-2} + \theta \bar{E}^{n-3}, \qquad 0 \le \theta \le 1. \tag{9}$$

This corresponds to the infinite impulse response filter shown in Fig. 4. Note that the response of the filter is near unity at low frequency, indicating that the filter passes the
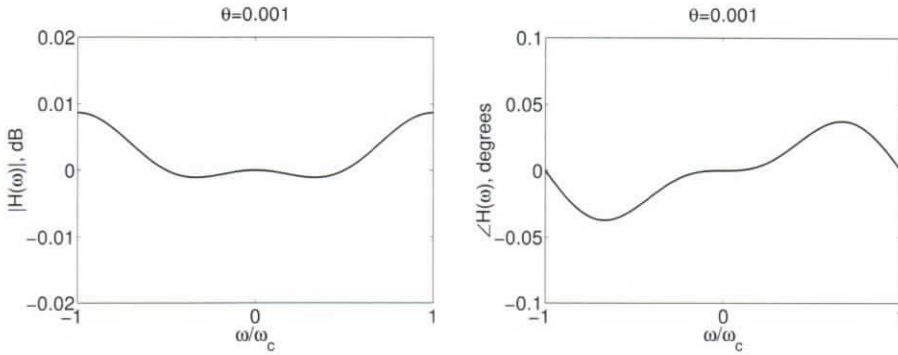
Fig. 4. Response of the Friedman filter with $\theta = 0.001$.
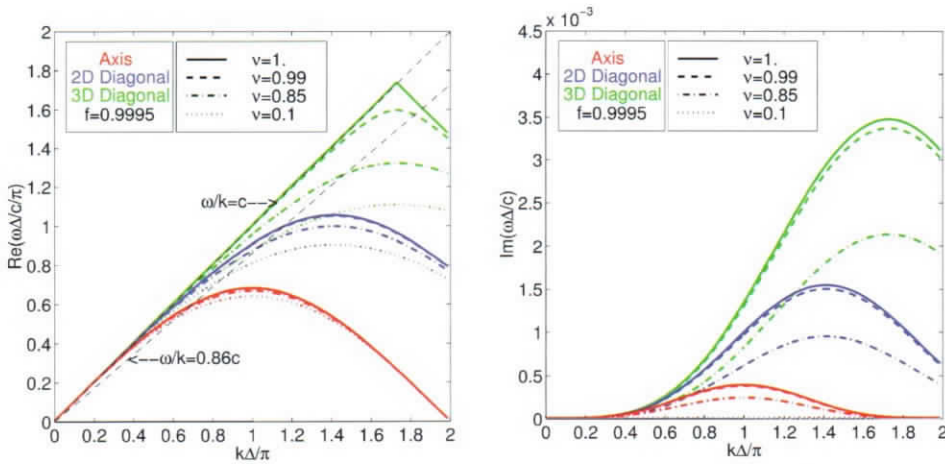

Fig. 5. Dispersion relation of the Friedman filter field solve with $\theta = 0.001$. The maximum stable timestep is decreased by a factor of $f = 0.9995$ from the maximum stable timestep of the Yee algorithm.

frequencies of physical interest without alteration. The magnitude of the filter response is greater than unity at high frequency, which increases the propagation speed of the waves that cause numerical Cerenkov radiation. The phase response of the filter also attenuates the high-frequency waves. Further, the Friedman filter is causal, meaning that no iterative scheme is required for its implementation. The features of this filter are seen in the dispersion relation plots of Fig. 5. Note that the maximum stable timestep is slightly lower than that of the Yee algorithm. In spite of this limitation, the lack of a requirement to iterate makes the use of this filter much more computationally efficient than the time-biased, iterative field solve algorithm.

Equation (7) is the central result of this effort to improve the ICEPIC numerical algorithms. The interpretation of the time-biased iterative field solve as a digital filter allows an understanding of how it eliminates numerical problems such as numerical Cerenkov radiation in PIC simulations. Equation (7) is then a vehicle for comparison to other methods of reducing or eliminating numerical problems in the simulations. Comparison between

the time-biased, iterative field solve and the field solve with the Friedman filter leads to the implementation of the Friedman filter in ICEPIC as the preferred method. Other avenues of eliminating numerical Cerenkov radiation are possible. For example, it is possible to replace the time filters in Eq. (7) with spatial filters, or it is possible to change the method of computing the spatial curl operator. However, the use of time filtering as presented here is the simplest method because spatial methods introduce the difficulty of handling boundary conditions near metallic surfaces and dielectric interfaces. Research efforts continually result in improvement to the numerical algorithms in ICEPIC.

6. ICEPIC Parallel Performance

The spatial resolution of a PIC simulation is fundamentally limited by the temperature and density of the charged particles, by the level of detail in the geometry of the device, and by the shortest wavelength that must be accurately modeled. For the devices of interest in this effort, these limitations dictate cell sizes on the order of 0.1–2.0 mm. The devices are typically approximately 30 cm in diameter and 100 cm in length. Simulations of such devices therefore require usually no fewer than 1 million cells and potentially as many as 1 billion cells. The timestep of a PIC simulation is limited by the time required for a wave to traverse a cell at the speed of light. Therefore, timesteps are generally in the picosecond range for cells with linear dimensions in the millimeter range. Simulations typically must compute several tens of nanoseconds of the device's behavior, thus requiring several tens of thousands of timesteps. On today's class of microprocessors operating with clock cycles of several hundreds of megahertz and with 0.25–2.0 GB of memory, we can perform a computational time advance of 500,000 field or particle data points in seconds of CPU time. Therefore, total run times for simulating realistic HPM sources are generally several days in length, and several scores of processors are required. Clearly, we require parallel computing platforms to carry out the numerical simulations with ICEPIC.

ICEPIC was designed with efficient operation on distributed memory parallel computers in mind. These include machines such as clusters of low-cost PCs and supercomputers such as the IBM SP3 or Cray T3E. These are the dominant HPC platforms today. ICEPIC was also designed to be portable so that it would be able to run on the HPC platforms of tomorrow.

The key characteristic of these machines is nonuniformity of memory access. Each CPU has some amount of local memory that can be accessed very quickly. However, the bulk of the memory on the machine (the memory not grouped with that particular CPU) can only be accessed very slowly compared to the local memory, and in many cases this memory access can only be done via communication channels between CPUs. Such communication is typically much slower than direct memory access by multiple orders of magnitude and also incurs additional communication overhead.

Efficient operation on such a machine requires that the bulk of computation by any CPU be on data in its local memory. Applied to our finite difference time domain (FDTD) method of solving Maxwell's equations and the Lorentz force equation for particles, this requirement yields a natural way of dividing up the problem among many CPUs. Updating the fields requires knowing the values of immediately neighboring fields and particles. Updating the particles requires knowing the values of immediately neighboring fields. Any partitioning of the entire task on many CPUs should keep adjacent fields and particles together in local memory as much as possible and minimize the amount of communication between CPUs. One way to achieve this is to cut the simulation region into subdomains and
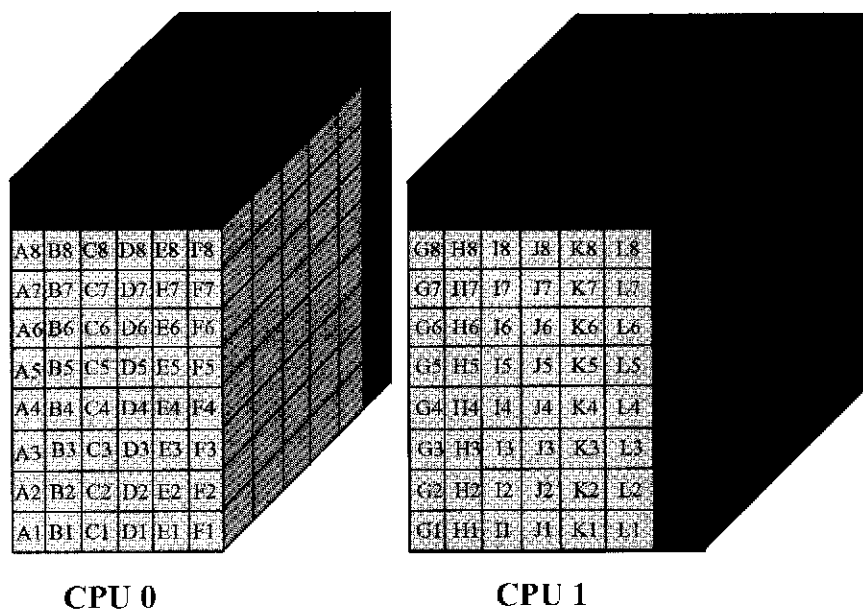
Fig. 6. Entire simulation domain divided into two pieces, with one piece assigned to each CPU, which is responsible for updating both cells and particles in its subdomain.

assign one subdomain to each CPU. Adjacent subdomains must exchange data in order to update fields and particles near their edges, and so in order to minimize communication one also wants to minimize the surface area of each subdomain. These considerations lead to a partitioning scheme such as that in Fig. 6, where a CPU is assigned a rectangular subdomain of the entire region and is responsible for updating the fields and particles therein. This method of dividing up the work is called "single domain decomposition." In "dual domain decomposition," particles and fields are partitioned separately to CPUs. Particles that cross from one subdomain to another during an update must also be transferred.

This type of partitioning ensures that most computation is done using only local data. However, any fields or particles near a shared subdomain boundary will still require communication to update, and this communication costs much time. Simply waiting for communication to finish before completing the update results in idle CPUs and wasted resources. ICEPIC addresses this problem by sending a message containing any information needed by neighboring CPUs before doing computation on local data. Then, after completing the updates on all local data that need no communication from neighbors, each CPU receives the needed data for edge updates and completes the update for that timestep. This style of messaging we call "asynchronous," since the process on the CPU receiving data does not necessarily receive them when the sending CPU sends them. Such a design works best when special communication hardware handles the messages without intervention of the CPU. Figure 7 shows the communication pattern ICEPIC actually uses. Let us consider the example of the "Send J" message. At that point, the CPUs sends the current (J) due to the particles in the border cells. Then, each CPU updates all the electric fields in its local memory that do not require these border J values, while the "Send J" message is in transit. When this computation has been completed, each CPU will either receive the J message immediately or wait until it arrives before proceeding to the update of the border electric
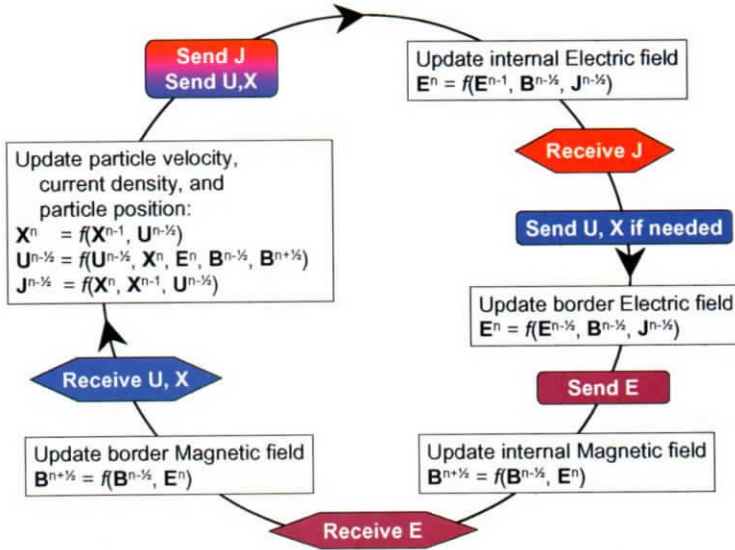
Fig. 7. ICEPIC performs computation on local, or internal, data while messages are sent from CPU to CPU.

field values. In this way, computation takes place during the relatively long period of time it takes for the J message to be sent from one CPU to another.

The requirement that neighboring CPUs send data to each other to complete an update causes an additional complication. Consider a case where there are two subdomains, one each on two CPUs, A and B. Suppose CPU A has 10 times more work than CPU B. It will soon happen that CPU B will be idle approximately 90% of the time because it cannot proceed without a message from CPU A, which is delayed because CPU A has not completed its own update. The best case is if every CPU has the same amount of work, a situation described as "load balanced." At program startup, partitioning can be done such that each CPU has an equal amount of work, and the load is very well balanced. This happy situation often does not persist since particles move about in ways we would not trouble to calculate if we could predict. ICEPIC not only incorporates automatic partitioning and load balancing at program start but also dynamically migrates work from CPU to CPU by moving cells and the particles in those cells. This equalizes the amount of work as the various CPUs become loaded, a process called "dynamic load balancing," which is performed at intervals of approximately 100 timesteps.

ICEPIC uses the following equation to perform both initial and dynamic load balancing:

$$W_{subdomain} = N_{cell} \times W_{cell} + N_{particle} \times W_{particle}, \tag{10}$$

where $W_{subdomain}$ is the total amount of work in the subdomain that contains N_{cell} number of cells and $N_{particle}$ number of particles, with a computational cost of W_{cell} per cell and $W_{particle}$ per particle. First, ICEPIC calculates the total amount of work in the simulation, and then it assigns cells and the particles contained in those cells to a subdomain, until that subdomain has approximately its equal share of the total work. However, ICEPIC restricts the shape of resulting subdomains to rectangular prisms and furthermore can only dynamically exchange cells in one dimension. These restrictions impose limits on the achievable efficiency resulting

from the dynamic load balancing. Stopping the simulation and restarting it will result in a well-balanced simulation, but this is inconvenient and not often done in practice.

These three features, asynchronous messaging, single-domain decomposition, and dynamic load balancing, are the methods built into ICEPIC for efficient use of today's HPC resources. In practice, the efficiency that ICEPIC obtains is highly problem dependent. For example, consider a case where you have a very small problem (10 MB) and divide it among 1,024 CPUs. Each CPU will have so little local work that it will spend most of its time waiting for messages to arrive, yielding terrible efficiency. It would be better to run such a problem on one or two CPUs, no matter how cleverly and efficiently ICEPIC is designed for parallel operation. The ability of a program such as ICEPIC to efficiently exploit many CPUs is often called the "scalability" of the code. There are several methods of measuring scalability. One is to start with a problem of a specific size and put it on more and more CPUs and see how much faster a solution is obtained. Another is to keep the amount of work per CPU constant but increase the size of the computation by a factor of the number of CPUs. Efficiency of the latter measure typically stays closer to 1.0 than the former because there just is not enough local computation to hide the cost of the communication time.

Figure 8 shows the scalability of ICEPIC on a simulation of the University of Michigan's MELBA relativistic magnetron microwave device.[17] These simulations were performed on a Compaq SC40 supercomputer belonging to the U.S. Army Engineer Research and Development Center, in Vicksburg, Mississippi. This simulation required approximately six million cells and was too large to fit on fewer than eight CPUs.

Figure 8 also shows ICEPIC's scalability on a scaled problem, where the number of cells per CPU is kept constant and the number of particles so low as to impose negligible computational cost. Here, we see good scalability up to 256 CPUs (the most we could
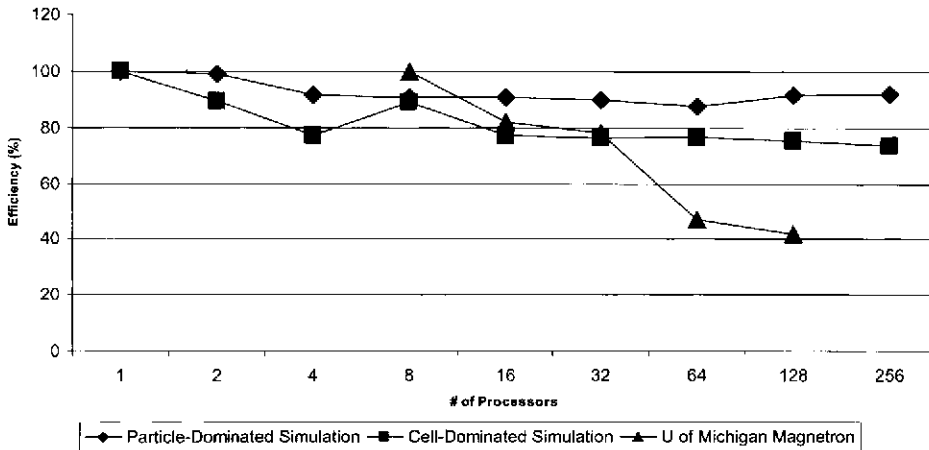

Fig. 8. University of Michigan magnetron line shows ICEPIC efficiency vs number of CPUs on a Compaq SC40 (833-MHz Alpha EV68 CPUs), for a fixed-size simulation of a magnetron. Efficiency drops to 50% at 64 CPUs but remains good for 16 and 32 CPUs. The cell-dominated simulations show good scaling when the load of cells per CPU is kept constant. The particle-dominated simulations illustrate good scaling when the number of particles/CPU is kept constant.

obtain). This result suggests that communication delays were playing a large part in the degradation of performance on the magnetron simulation described previously. Figure 8 also shows ICEPIC's scalability when the number of particles per CPU is kept constant and the number of cells is kept so low as to impose negligible cost.

From these plots we can see that ICEPIC is not only scalable on problems designed purely for testing scalability but also on real-world problems such as simulating the Michigan magnetron experiment.

7. ICEPIC Application: MILO

Systematic improvements in the electrostatic design of MILO velvet cathodes have led to corresponding gains in tube performance.[12, 15] In particular, the RF pulse duration at full power has increased from 75 ns (Ref. 6) to over 400 ns (Ref. 15).

The most vital and troublesome portion of MILO cathode design is at the ends of the cathode (see Fig. 9 for the tube geometry). Problems due to edge effects are especially harmful at the launch point for the current flow. A dc impedance discontinuity that acts as a current launcher[18] exists between the third choke vane and the downstream slow-wave structure. This is a key element in increasing the efficiency of RF power production for MILO.[16] (Note that the tapered MILO[8] uses a different scheme to enhance tube efficiency.) However, this beneficial impedance discontinuity comes at the expense of a very high dc electric field (more than 300 kV/cm) and resultant high beam current density (more than 300 A/cm^2) at the launch point. This can lead to anode plasma formation[7] that severely perturbs the launch point flow, as well as causing cathode flares[19] that result in asymmetric emission from the cathode and physical damage to the slow-wave structure.[14]

We will show the consequences of anode plasma formation in the MILO. Two-dimensional PIC simulations including self-consistent anode plasma formation show that
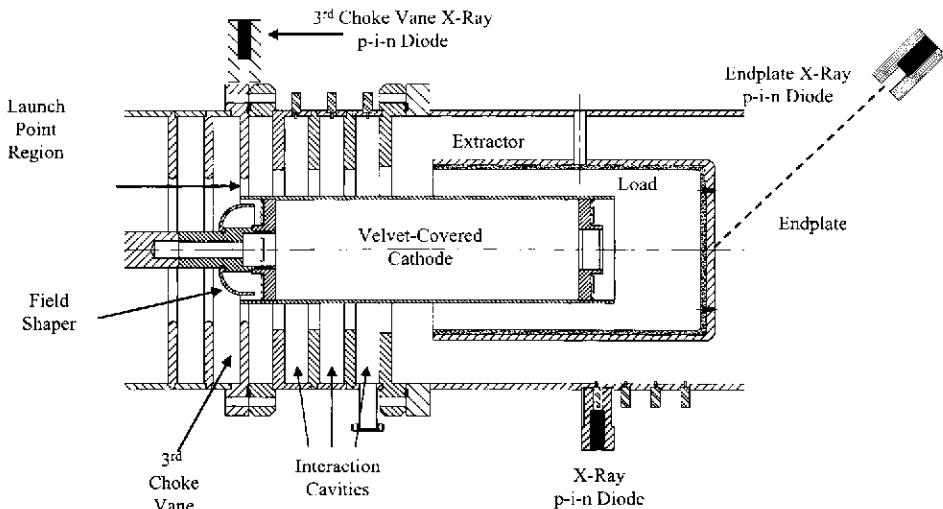

Fig. 9. Diagram of the MILO geometry with the field-shaper cathode showing the lead-collimated x-ray p-i-n diode placement.
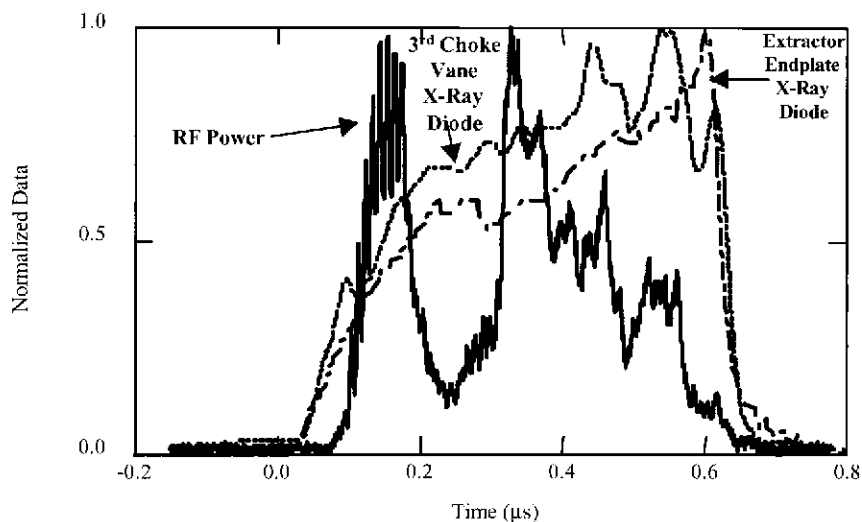
Fig. 10. Overlay of radiated microwave power (——) and x-ray p-i-n diode signals from the third choke vane (· · ·) and the extractor endplate (·-·) for the case of the field-shaper cathode.

the first two interaction cavities are effectively shorted out, leading to a power dip virtually identical to the one observed in the experiment.[15]

The radiated output power from the MILO is shown in Fig. 10, along with overlays of x-ray emission from the third choke vane and from the extractor endplate. Several interesting features are evident in these data, the most striking of which is the large power dip that occurs in the middle of the microwave pulse. The fact that the power is able to completely recover in amplitude before again tapering off was most puzzling. Previous computer simulations, where bipolar space-charge flow[14] was enabled (but not self-consistently) on the third choke and in the load region,[15] were able to explain the x-ray data, the lower-than-expected dc impedance and RF efficiency, and the frequency spectrum but not the power dip and subsequent recovery. Thus, a new set of 2D simulations using ICEPIC[20] (which agree with OOPIC[26] simulations not shown here) were undertaken to investigate whether self-consistent plasma formation in the tube and ensuing bipolar space-charge flow could replicate the experimental results.

Despite attempts in the laboratory to keep background gas densities low, neutral gas emanates from plasma-facing surfaces via physical processes such as thermal desorption and stimulated desorption. Electrons and ions that collide with background neutral gas molecules can ionize them. Electrons that collide with solid walls can produce secondary electrons. Both types of collision events have the potential to modify the plasma density and will influence the physical behavior of the device under investigation. Hence, for more realistic simulations, we have begun to develop numerical models for both of these time-dependent plasma formation processes in enclosed structures.

In modeling this device, we simulated the MILO, with the addition of hydrogen gas of constant pressure filling the tube. (The validity of this assumption is discussed later in this section.) The following physical processes were performed self-consistently with ICEPIC: the primary beam electrons strike the anode during the preinsulation phase, producing
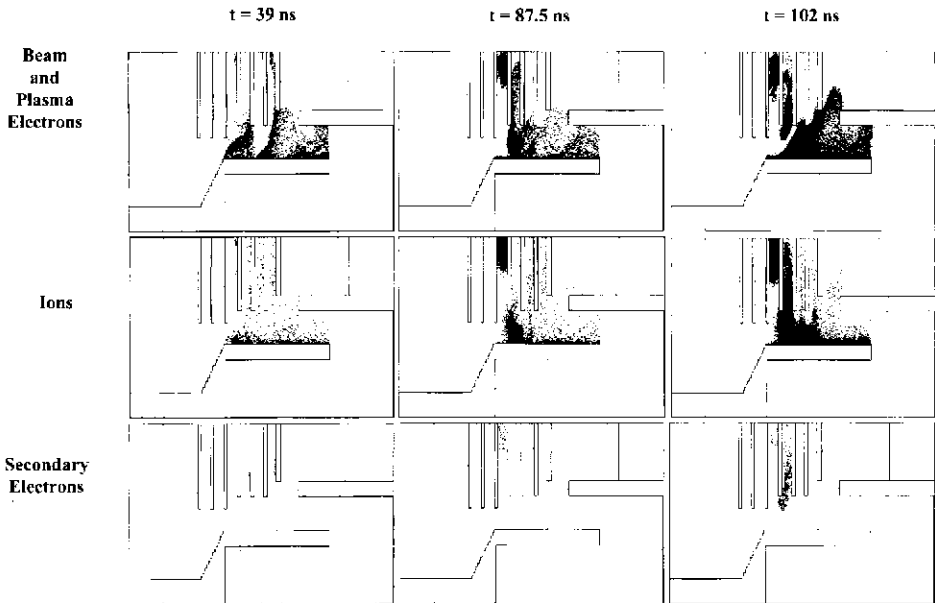
Fig. 11. ICEPIC simulation results showing snapshots of particle trajectories (beam and plasma electrons along the first row, ions along the second row, and secondary electrons along the third row) at three different times ($t = 39$ ns along the first column, $t = 87.5$ ns along the second column, and $t = 102$ ns along the third column).

secondary electrons that generate hydrogen plasma via impact ionization with the neutral gas. Ion-neutral and electron-neutral impact ionization is done using the null collision method developed by Vahedi and Surendra.[24] Note that the ion-neutral ionization dominates the electron-neutral ionization for higher energies (above 4 keV). The highest plasma density occurs in the launch-point region because it has the highest current density in the cathode. Therefore, the third choke vane has the largest secondary yield for participating in impact ionization. The secondary electron model (based on previous work by Vaughan,[25] Shih,[23] and Gopinath[11]) for the secondary electron emission coefficient δ depends on the energy and angle of incidence of the primary electron.

Once the plasma density is large enough, bipolar space-charge flow initiates. The consequences of this are seen in Fig. 11, which shows snapshots of particle trajectories (beam electrons, plasma electrons, ions, and secondary electrons) at three different times in the simulation. At 39 ns into the voltage ramp, the spokes are well developed and RF generation is near peak amplitude. Secondary electrons are seen to form when spoke electrons strike the vane tips. At 87.5 ns, space-charge flow is apparent underneath the fourth vane. In addition, plasma has filled the top half of the first interaction cavity (the third cavity from the left end of the slow-wave structure). These two effects completely disrupt the launch-point flow, and spoke formation essentially ceases. This leads to a large drop-off in microwave power. At time 102 ns, for reasons not completely understood, the region of intense bipolar flow partially disassembles, allowing spoke formation to resume. Note, however, that only the last two interaction cavities now participate in meaningful RF production. Nevertheless, the amplitude of the RF power completely recovers for a brief period before again falling off in
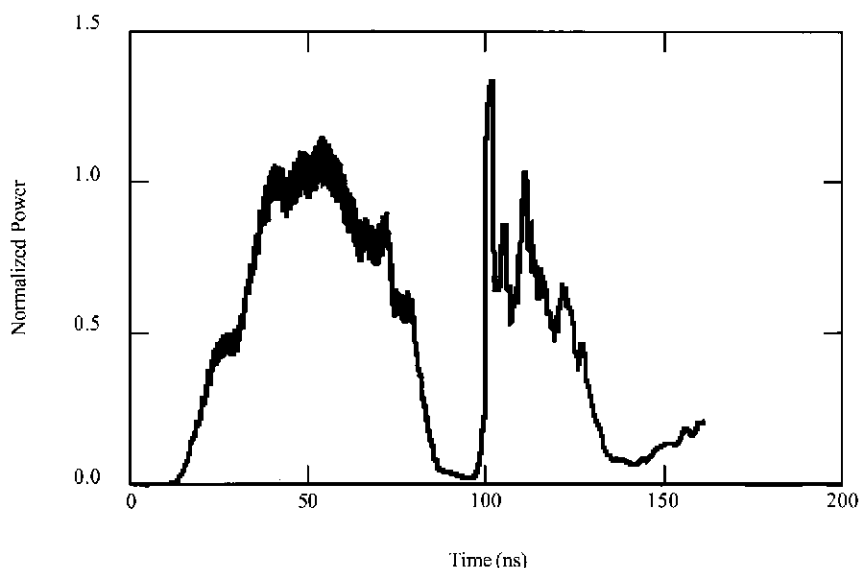
Fig. 12. ICEPIC RF output power prediction. Same normalization is used on the simulated power as was used on the experimental power shown in Fig. 10.

amplitude. The overall effect on the output power is shown in Fig. 12, which has features remarkably similar to the experimental data in Fig. 10.

The experimental results indicate that the ICEPIC model overestimates the amount of plasma produced in MILO. This is consistent with the fact that the durations of both the power dip and the recovery phase are somewhat shorter in the simulations than observed in the experiment. Of course, the background gas aspect of the ICEPIC simulations is a highly idealized model of the experimental conditions. The laboratory MILO has a base vacuum of 1×10^{-6} torr and does not have an initial gas fill as was used in the simulations. The most likely scenario is that a gas layer forms on the anode surface due to stimulated desorption[7] by beam electrons. On the time scale of the beam pulse, this neutral gas layer does not move and is ionized by secondary electrons also generated at the anode surface. In future simulations we plan to incorporate stimulated desorption into the anode physics model, but as initial models, the ICEPIC simulations reported here have worked quite well.

In conclusion, the dip in power is caused by plasma formation. In general, to accurately simulate the power output of an HPM tube the surface and neutral gas physics needs to be accounted for. With a relatively simple surface model, we have been able to have qualitative agreement with experiment.

8. Concluding Remarks

ICEPIC is being used to design RF weapons to work against hostile battlefield electronics. Given the importance of electronics in the modern battlefield, the ability to target these electronics will give the United States a distinct advantage against technically advanced adversaries. The effect of RF radiation on the electronics can vary from transient disruption to destruction of the hardware, depending on the power and frequency of the radiation.

Furthermore, since only electronic equipment is affected, RF radiation can be applied in a nonlethal manner, giving the war fighter a broader range of options for dealing with opponents of various degrees of sophistication and strength. The use of ICEPIC to design sources of electromagnetic radiation is critical to the DoD's advanced RF weapon effort.

At present the state of the art in computational power and algorithm accuracy allow us to simulate devices tens of wavelengths on a side. We are continuing to improve our parallel performance so that we can run on more and more processors. We are also looking into better algorithms so that we can simulate larger problems. Currently, we can simulate from the pulse power through to the antenna; however, we would like to be able to include surrounding structures as well, to predict electromagnetic interference and electromagnetic compatibility.

9. Acknowledgments

References

[1] Arman, M.J., IEEE Trans. Plasma Sci. **24**, 964 (1996).

[2] Benford, J., and L. Myrabo, Proc. SPIE **2154**, 198 (1994).

[3] Benford, J., and J. Swegle, *High-Power Microwaves*, Artech House, Boston (1992).

[4] Birdsall, C.K., and A.B. Langdon, *Plasma Physics via Computer Simulation*, McGraw–Hill, New York (1985).

[5] Boris, J.P., "Relativistic Plasma Simulation-Optimization of a Hybrid Code," Proc. Fourth Conf. Num. Sim. Plasmas, Naval Res. Lab., Washington, DC, pp. 3–67 (1970).

[6] Calico, S.E., M.C. Clark, R.W. Lemke, and M.C. Scott, Proc. SPIE **2557**, 50 (1995).

[7] Cuneo, M.E., P.R. Menge, D.L. Hanson, W.E. Fowler, M.A. Bernard, G.R. Zisks, A.B. Filuk, T.D. Pointon, R.A. Vesey, D.R. Welch, J.E. Bailey, M.P. Desjarlais, T.R. Lockner, T.A. Mehlhorn, S.A. Slutz, and M.A. Stark, IEEE Trans. Plasma Sci. **25**, 229 (1997).

[8] Eastwood, J.W., K.C. Hawkins, and M. P. Hook, IEEE Trans. Plasma Sci. **26**, 698 (1998).

[9] Friedman, A., US–Japan Workshop on Advanced Computer Simulation Techniques Applied to Plasmas and Fusion (1990).

[10] Godfrey, B.B., and B. Goplen, "Practical Evaluation of Time-Biased Electromagnetic Field Algorithms for Plasma Simulations," Mission Research Corporation Report, AMRC-N-146 (1980).

[11] Gopinath, V.P., and B.H. Vanderberg, Phys. Plasmas **5**(1), 261 (1998).

[12] Haworth, M., G. Baca, J. Benford, T. Englert, K. Hackett, K. Hendricks, D. Henley, M. LaCour, R. Lemke, D. Price, D. Ralph, M. Sena, D. Shiffler, and T. Spencer, IEEE Trans. Plasma Sci. **26**, 312 (1998).

[13]Haworth, M.D., K.L. Cartwright, J.W. Luginsland, D.A. Shiffler, and R.J. Umstattd, IEEE Trans. Plasma Sci. **30**, 992 (2002).

[14]Haworth, M.D., J.W. Luginsland, and R.W. Lemke, IEEE Trans. Plasma Sci. **28**, 511 (2000).

[15]Haworth, M.D., J.W. Luginsland, and R.W. Lemke, IEEE Trans. Plasma Sci. **29**, 388 (2001).

[16]Lemke, R.W., S.E. Calico, and M.C. Clark, IEEE Trans. Plasma Sci. **25**, 364 (1997).

[17]Lopez, M.R., R.M. Gilgenbach, D.W. Jordan, S.A. Anderson, M.D. Johnaton, M.W. Keyser, H. Miyake, C.W. Peters, M.C. Jones, V.B. Neculaes, Y.Y. Lau, T.A. Spencer, J.W. Luginsland, M.D. Haworth, T.W. Lemke, and D. Price, IEEE Trans. Plasma Sci. **30**, 947 (2002).

[18]Mendel, C.W., M.E. Savage, D.M. Zagar, W.W. Simpson, T.W. Grasser, and J.P. Quintenz, J. Appl. Phys. **71**, 3731 (1992).

[19]Messyats, G.A., and D.I. Proskurovsky, *Pulsed Electrical Discharge in Vacuum*, Springer, Berlin (1989).

[20]Peterkin, R.E., and J.W. Luginsland, Comp. Sci. Eng. **4**(2), 42 (2002).

[21]Rambo, P., J. Ambrosiano, M. Friedman, and J.D.E. Nelson, 13th Conference on the Numerical Simulation of Plasmas (1989).

[22]Sasser, G.E., L. Bowers, S. Collela, D. Lileikis, J. Luginsland, D. McGrath, L. Merkle, R.E. Peterkin, and J.J. Watrous, "Virtual Prototyping of Microwave Devices using MHD, PIC, and CEM Codes," AIAA Paper 98-2926 (1998).

[23]Shih, A., and C. Hor, IEEE Trans. Electron. Dev. **40**(4), 1963 (1993).

[24]Vahedi, V., and M. Surendra, Comp. Phys. Comm. **87** (1–2), 179 (1995).

[25]Vaughan, J.R.M., IEEE Trans. Electron. Dev. **36**(9), 1448 (1989).

[26]Verboncoeur, J.P., A.B. Langdon, and N.T. Gladd, Comp. Phys. Comm., **87**, 199 (1995).

[27]Villasenor, J., and O. Buneman, Comp. Phys. Commun. **69**, 306 (1992).

[28]Yee, K.S., IEEE Trans. Ant. Prop. **AP-14**, 302 (1966).

The Authors

Keith L. Cartwright received his B.S. degree in physics from University of Illinois at Champaign–Urbana, in 1991, and his M.S. and Ph.D degree in physics from the University of California, Berkeley, in 1999. His graduate work involved the development of particle-in-cell (PIC) algorithms and PIC simulation of nonneutral and quasineutral plasmas and microwave devices. He joined the Directed Energy Directorate of the Air Force Research Laboratory, Kirtland AFB, NM, in 2000. His research interests include simulation and theory of high-power microwave devices.

Andrew D. Greenwood received the B.S. degree in 1993 and the M.S. degree in 1995, both from Brigham Young University in electrical engineering. He received the Ph.D. in electrical engineering in 1998 from the University of Illinois, where he studied the use of the finite element method to compute electromagnetic scattering and radiation from axisymmetric bodies. In 1996, he joined Rome Laboratory as a Palace Knight employee, and in 1998 he joined the Directed Energy Directorate of the Air Force Research Laboratory at Kirtland AFB, NM, where he conducts research on particle-in-cell methods to simulate high-power microwave devices.

Michael D. Haworth received the B.S. degree in physics in 1977 and the M.S. and Ph.D. degrees in plasma physics in 1978 and 1983, respectively, from Auburn University, Auburn, AL. From 1983 to 1992, he worked for Science Applications International Corp., Albuquerque, NM, in the fields of plasma guiding of intense relativistic electron beams and virtual cathode oscillators. He then served as section leader for the drift tube linac at the Superconducting Super Collider Laboratory from 1992 to 1994. In 1994, he joined the Directed Energy Directorate of the Air Force Research Laboratory, Kirtland AFB, NM. At present, he is a senior research physicist in the High Power Microwave Division.

Lt Tony A. Murphy enlisted in the U.S. Air Force on June 23, 1988, and served as an F-16 Avionics Repairman for 10 years. As enlisted he was stationed at Nellis AFB, NV

(1989–1993); Osan AB, ROK (1993–1994); and Eglin AFB, FL (1994–1998). In 1998, Lt Murphy was accepted into the Airman's Education and Training Program, to pursue a bachelor's degree in engineering. On December 16, 2000, he earned dual Bachelor's degrees in Electrical Engineering and Computer Engineering with High Honors from the University of Florida at Gainesville. On April 5, 2001, Lt Murphy received a commission as a 2nd Lieutenant in the U.S. Air Force at Officer Training School, Maxwell AFB, AL. He is now stationed at the Air Force Research Laboratory, Directed Energy Directorate on Kirtland AFB, NM. Lt Murphy is currently a computer engineer on the computational physic team.