

.....  
University of California - Berkeley

# Advanced Microwave Tube Computer Experiments Research

**F49620-96-1-0154**

*Final Progress Report*

*1 May 1996 – 30 April 2000*

**Submitted to the**

**Air Force Office of Scientific Research**

**by**

**Professor Charles K. Birdsall**

**Department of Electrical Engineering & Computer Science**

# 1. INTRODUCTION

## 1.1 Scope

This report summarizes the progress of the Plasma Theory and Simulation Group (PTSG) under the Western Consortium of the Multidisciplinary University Research Initiative (MURI) High Energy Microwave (HEM) research program. The PTSG, in the Department of Electrical Engineering and Computer Sciences at the University of California in Berkeley, is a member of the MURI-West consortium lead by the University of California at Davis.

This report covers the period between August 1, 1999 through March 14, 2000. Detailed technical information on the research projects described is not presented here, but can be found in attached or pending journal publications, reports, and conference papers.

The PTSG is primarily involved in the modeling of microwave-beam, plasma, and vacuum electronics devices, using the tools of theory and simulation. Here, plasma is defined in the broad sense to include non-neutral plasmas and electron beams. The PTSG also develops, releases, and supports a number of plasma simulation codes. The codes are in use by hundreds of researchers worldwide in academia, national laboratories, and industry, as well as across MURI. The PTSG suite of codes is available via the Internet at <http://ptsg.eecs.berkeley.edu>.

## 1.2 PTSG Members

The Plasma Theory and Simulation Group at the University of California, Berkeley currently consists of one professor, two research engineers, two postdoctoral researchers, one visiting faculty, and five Ph.D. students, one graduate student on leave, as well as a varying number of industrial and academic visitors. The PTSG members supported by the MURI project, and their approximate MURI-funded support levels, under the term covered by this report are:

- Prof. C. K. Birdsall, principal investigator [5%]
- E. Kawamura, Ph.D. student [100%]
- P. J. Mardahl, Ph.D. student [100%]
- J. M. Oslake, Ph.D. student on leave [5%]
- Dr. J. P. Verboncoeur, research engineer [33%]

During this reporting period, PTSG admitted two new Ph.D. students, Xingbo Yu and Jason Dimkoff. One Ph.D. student (J. M. Oslake) remains on a leave of absence, but is continuing to work on completing a publication on his eigensolver research. In addition, K. L. Cartwright, not presently funded by this MURI, provided a significant benefit to MURI through enhancements to XOOPIE and interaction with AFRL personnel; Dr. Cartwright graduated during this period and accepted a position at the AFRL in Albuquerque.

## 1.3 Outline of Report

This report is organized topically in sections as follows. In Section 1, the scope of the document, personnel working in this area, and the outline of the report are described. In Section 2, the progress on the research projects is described for the period covered by this report. In

## 2. RESEARCH PROJECTS

This section describes the progress on research projects at the University of California, Berkeley, branch of the MURI-West High Energy Microwave Sources consortium. Note that some of these projects are primarily funded by other agencies (the percentage of MURI support is noted when less than 100%), and are included here due to their relevance to HEM. Also note that some projects are complete, as described below.

### 2.1 XOOPIC

This section summarizes the XOOPIC code, described in part in the literature<sup>1</sup>. The emphasis of the work in this reporting period was the parallel and three-dimensional extensions of the XOOPIC code.

#### 2.1.1 Background

The initial release of XOOPIC, version 1.0, was presented at the OOPIC Release Workshop, University of California at Berkeley, CA, September 1995. Version 1.1 was released just before IEEE ICOPS, in June, 1996.

The OOPIC project started in October 1993 as an AFOSR-funded joint effort between University of California, Berkeley (physics) and George Mason University (graphics), with industrial participants Berkeley Research Associates (expert systems) and FM Technologies (documentation, administration). The present version runs on Unix workstations, using the University of California, Berkeley XGrafix visualization system. UC-Berkeley is currently the primary active code developer.

As one of the pioneering efforts in object-oriented (OO) scientific programming, XOOPIC has demonstrated conclusively the benefits of the method. The code extensibility and development costs are lower, and the development more rapid than traditional methods. The OO technique is used in most of the XOOPIC development work. Similar methods of simplifying development and maintenance are now propagating into industrial and laboratory codes, including MRC's MAGIC and AF Phillips Lab's ICE-PIC. The specific benefit for HEM (and other DOD) objectives is a more reliable, freely distributed tool set. Eventually there will be interchangeable parts (models) between separate codes, which should allow more rapid modeling of devices with reduced development costs.

XOOPIC was originally intended to be a two-dimensional electromagnetic PIC code for modeling microwave-beam devices. Due to flexibility of the underlying architecture, it has grown to encompass a significantly larger range of plasma and beam devices. The code has been used to simulate devices including relativistic klystrons, Cerenkov masers, low and high pressure discharges, and beam optics with and without plasma. In the sections below, the capabilities of each version of the code are described.

- The most significant advances made during the reporting period include a parallel computing capability and the design and implementation for extension to three dimensions. These advances are described in detail below. For the details of the previous versions of XOOPIC, refer to previous reports.

## **2.1.2 Parallel PIC**

This section describes the parallel extension of the XOOPIE particle code. The parallel capability provides the necessary computing power for application to three dimensional problems. Parallel XOOPIE currently runs on heterogeneous workstations distributed on standard ethernet-type networks, networks of workstations (with specialized communications linkages, called NOWs), as well as massively parallel platforms and clusters of symmetric multiprocessing (SMP) computers. This research project has significant impact on the HEM mission by providing a portable tool to rapidly model microwave devices, including kinetic effects, without any mode constraints. Significant collaboration is ongoing on this topic with the ICEPIC development effort at AFRL.

The design and implementation have evolved to better accommodate the development of the three-dimensional version. In addition, new features have been added, particularly in creating parallel versions of boundary conditions. In the sections below, a document being compiled on this topic for journal publication is presented.

### **2.1.2.1 Objectives**

The PIC method of simulating HPM devices is very CPU intensive. Simple devices can be simulated in minutes or hours: but complex devices can take weeks or months to simulate in sufficient detail to be useful. Parallelization can reduce the run-time of large simulations, allowing greater use of these codes in design efforts. Larger problems can also be simulated using a parallel code than a single-processor code: memory demands can be distributed across many nodes.

One particular example of a parallel PIC code is XOOPIE, a 2-d, 3-v electromagnetic plasma simulation program. XOOPIE has been successful as a single-processor code, and is able to simulate many interesting devices including relativistic klystron oscillators, electron guns, DC dischargers with gas chemistry, plasma display panel cells, and highly relativistic beams in accelerators. XOOPIE is written in C++ and uses the MPI library for parallel communication, which allows the code to be used on a broad range of parallel hardware, including clusters of workstations, SMP machines, and massively parallel machines. The parallel version of XOOPIE leverages much of the work done on the single-processor version: it has most of the same capabilities as well as being able to use multiple CPUs on the same problem. The source code of XOOPIE is public domain, so the interested reader can obtain the code to understand its workings, or modify it to a particular purpose.

### **2.1.2.2 Strategy**

The strategy for parallelization of XOOPIE is a coarse-grained spatial decomposition of the physical model into computational regions, as shown in Figure 1. Each computational region has its own mathematical mesh and particle arrays. A coarse-grained partitioning as shown has advantages over other partitioning strategies: in order to update the electromagnetic fields on the mesh points, it is necessary to know the fields on neighboring mesh points. If the neighboring mesh points are on other CPUs, communication is required between CPUs, and typically, communication is much slower than computation on a parallel machine. In a block-cyclic decomposition, for example, many more mesh points would have to have non-local data communicated in order to update fields, than would with a coarse-grained partitioning.

## A sample partitioning

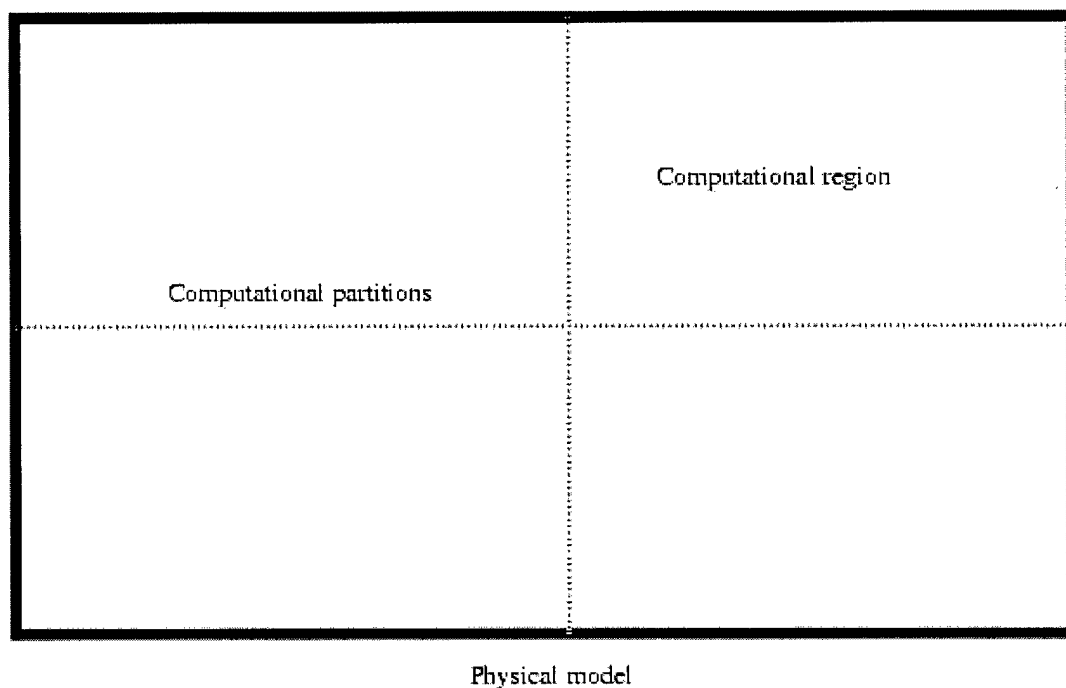


Figure 1. A sample partitioning scheme.

Another advantage of coarse-grained partitioning is re-use of code. We treat boundaries of the computational regions as boundary conditions. Therefore, the special case of updating the fields on a computational boundary can be encapsulated in a new boundary condition called the SpatialRegionBoundary (SRB), minimizing the number of modifications to the existing, already tested non-parallel version of XOOPIC.

Coarse-grained partitioning also allows ready identification of mesh points with only local dependencies: i.e., mesh points which require no data from remote CPUs to update. These “interior” mesh points can be updated while data from other CPUs is transmitted, allowing useful work to be done while messages are in transit. This is an important optimization to perform if parallel resources are to be used efficiently, and has been done in parallel XOOPIC, as shown in Figure 2.

Slow implementation      Faster implementation

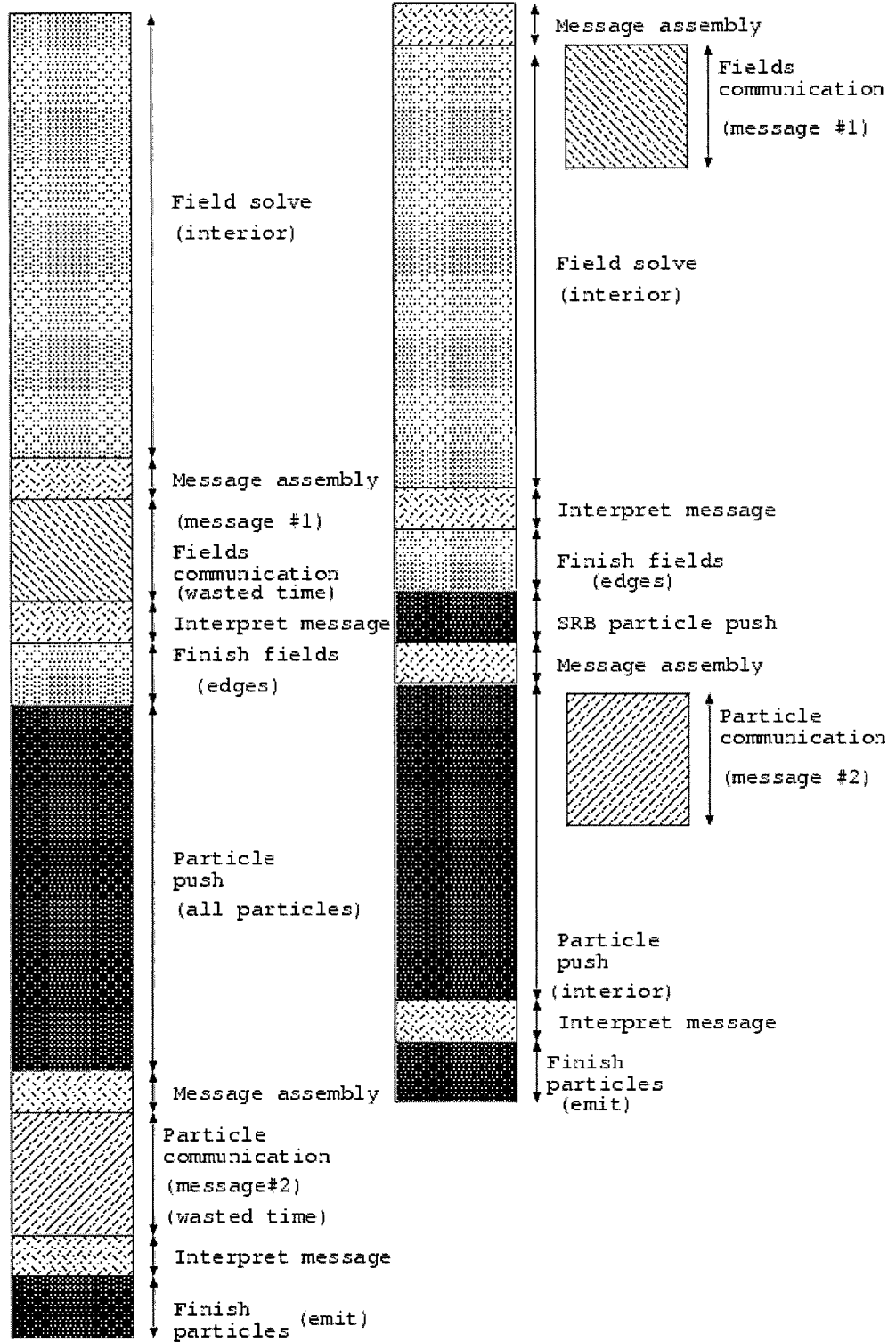


Figure 2. Hiding communication time behind local computation: computation can be done concurrently with message transmission.

Particles also require field data in order to update their positions. Communicating field data every time a particle needed it would be a fatal mistake if performance were a consideration. However, particles only need field data from adjacent mesh points, so if particles are assigned to the same CPU on which the fields it needs reside, communication is not required to update their positions. The exception to this is when particles cross from one region to another: in this case, the particle data is communicated to the destination CPU. In XOOPIIC, particles are distributed this way, so that the fields are nearly always local, but it is not possible to easily identify which particles will cross an SRB. This is unfortunate, because it makes another optimization difficult: if crossing particles could be identified early enough, they could be moved first, communicated to their destination, and while the data was in transit, particles needing only local fields could be updated (see Faster implementation, Fig. 2).

XOOPIIC at present uses the "Faster implementation" for the fields, and the "Slower implementation" for the particles.

Coarse-grained partitioning strategies lend themselves to the use of parallel libraries such as MPI (Message Passing Interface) or PVM (Parallel Virtual Machine). The MPI library in particular is widely portable and gives good performance, and we have used it for parallel XOOPIIC. Other tools for parallel programming exist, such as split-C and High Performance Fortran, but these require special compilers which are often not freely available on platforms of interest. Use of the split-C and HPF compilers would also require extensive modifications to the XOOPIIC code.

### ***2.1.2.3 Spatial Region Boundaries***

As remarked above, we encapsulated much of the work in parallelizing XOOPIIC in a special boundary condition: the Spatial Region Boundary (SRB). Figure 3 shows the fields on the mesh near and on an SRB. SRBs are created in linked pairs, with each virtual boundary (a boundary defining a computational region, not a physical boundary) requiring two SRBs, one per computational region. Each SRB is responsible for sending and receiving the necessary fields and passing and receiving particles which cross them.

In order to compute the field components correctly on the SRB (the fields in the dotted box), the field components external to the region must be communicated to it (the fields 1E1, 1E2, 1E3, 1B1, 1B3, and the currents, J1 and J3, which are in the same locations as E1 and E3 respectively).

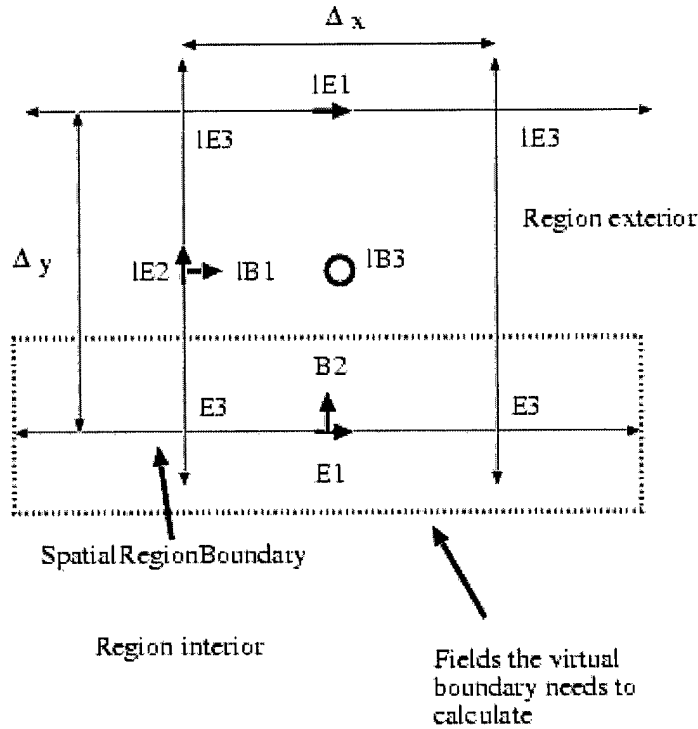


Figure 3. Fields on and near a horizontal Spatial Region Boundary.

The field solve on the boundary is actually a three-step process: IE1, IE2, and IE3 from the region exterior are used to update IB1 and IB3, which are stored in the SRB. E1 and E3 on the SRB are then computed using the updated ghost values of IB1 and IB3. B2 on the SRB is updated using the updated E3s on the SRB by the interior field solve, so it is only necessary for the SRB to set E3 properly. These are the computations performed by the SRB shown in Figure 3:

$$IB1_{jk}^{t+1/2} = IB1_{jk}^t - \frac{\Delta t}{2\Delta y} (IE3_{jk}^t - E3_{jk}^t)$$

$$IB3_{jk}^{t+1/2} = IB3_{jk}^t - \frac{\Delta t}{2} \left( \frac{E1_{jk}^t - IE1_{jk}^t}{\Delta y} + \frac{IE2_{j+1,k}^t - IE1_{jk}^t}{\Delta x} \right)$$

$$E1_{jk}^{t+1} = E1_{jk}^t + \frac{\Delta t}{\varepsilon} \left( \frac{IB3_{jk}^{t+1/2} - B3_{j,k-1}^{t+1/2}}{\mu\Delta y} - J1^{t+1/2} \right)$$

$$E3_{jk}^{t+1} = E3_{jk}^t + \frac{\Delta t}{\epsilon} \left( \frac{B1_{jk-1}^{t+1/2} - B1_{j,k}^{t+1/2}}{\mu\Delta y} + \frac{B2_{jk}^{t+1/2} - B2_{j-1,k}^{t+1/2}}{\mu\Delta x} - J3_{jk}^{t+1/2} \right),$$

where  $J1$  and  $J3$  are current densities from particle motions,  $\Delta t$  is the simulation time step, and the subscripts  $j, k$  are mesh point indices. The update of  $B$  is split into two phases, a half-update of  $B^t$  to  $B^{t+1/2}$ , an update of  $E^t$  to  $E^{t+1}$ , and another half-update of  $B^{t+1/2}$  to (not shown, because the non-parallel EM field solve does this properly when the internal fields are updated): this scheme achieves 2nd order accuracy for the field solve, and makes  $E^t$  and  $B^t$  available for the particle update. This is the reason for the use of  $\Delta t/2$  in the update of  $B$ . Note that the both of the paired SRBs are redundantly calculating the fields on the SRB itself, to avoid the necessity of having to communicate the result before moving particles.

XOOPIC tracks the trajectories of particles, and when particles cross boundaries, they are removed from the simulation and given to the boundary for disposition. Conducting boundaries simply destroy the particles, and dielectric boundaries may collect the charge of the particles at the point of impact. SRBs, when given particles, transfer them to the paired SRB on the CPU which is handling the adjacent computational region, and complete the particle update, so that the particle motion continues unperturbed through the SRB.

Figure 4 shows a flow diagram for the XOOPIC code which details when the SRBs send and receive messages. In step 1, XOOPIC updates all the particle positions. Particles which are crossing virtual boundaries are identified by the fact that they intersect an SRB. At the end of this stage, when all the particles have been moved, the SRB sends crossing particles to its counterpart SRB (msg \#2). When this message arrives (the dashed arrow), the SRB places any particles sent to it into the simulation (step 2.) Also in step 2, boundaries which emit particles (such as secondaries, a thermionic cathode, or a beam of particles) place their particles in the simulation.

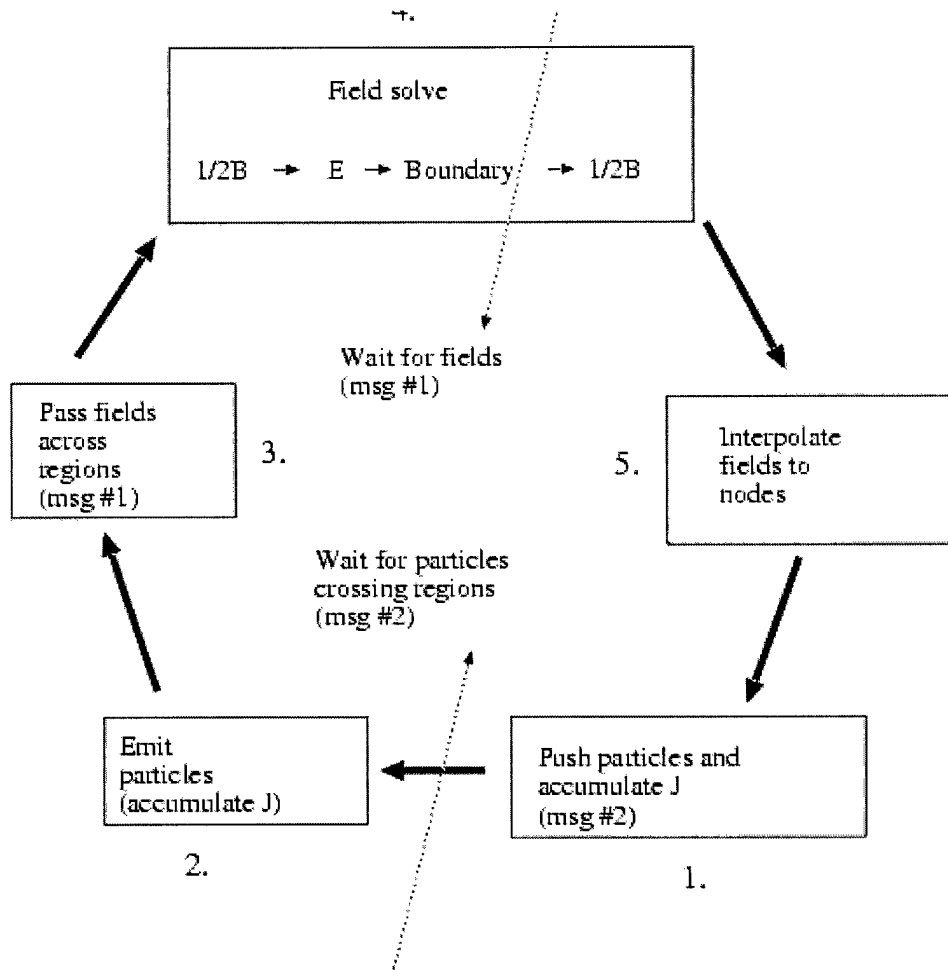


Figure 4. Flow diagram for parallel XOOPIC

When step 2 is completed, the field solve may begin. The current density,  $J$ , is required for the field solve, and cannot be computed until all particle positions are updated: it is when the particle positions are updated that the current due to particle motion is calculated. The first step of the field solve (step 3) is for the SRBs to communicate fields to their counterparts. A message is sent (msg #1) which contains the necessary field components. Computation is immediately begun on updating fields interior to the computational region, which proceeds while msg #1 is in transit, since those fields do not depend on external data. If computation reaches the "Boundary" stage before msg #1 has arrived, execution will halt until it arrives. Whether time is wasted or not depends on the comparison of the time it takes to update all interior points ( $T$ ) and the time it takes to transmit and receive the message ( $t$ ). If  $t > T$ , time is wasted.  $T$  and  $t$  are both highly dependent on the specific CPU and parallel architecture, and on the ratio of boundary points to interior points. In general, performance is best when there are many more interior points than boundary points.

#### 2.1.2.4 Parallel Poisson Solve

In an electrostatic solve the electrostatic potential at any mesh point depends on the charge and boundary conditions of the entire system, unlike the electromagnetic field solve, when the fields could be updated with information from only neighboring cells. Figure 5 shows the location on the mesh for the electrostatic quantities.

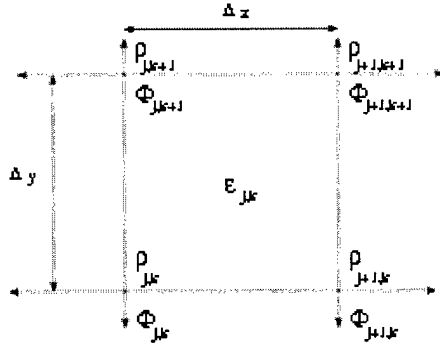


Figure 5. The grid locations of the electrostatic quantities.

In XOOPIC, a linear system of equations for solving for potential on the mesh is set up using the following discretization:

$$\nabla^2 \Phi \approx \frac{\Phi_{j,k+1}}{\Delta y^2} - \frac{2\Phi_{j,k}}{\Delta y^2} + \frac{\Phi_{j,k-1}}{\Delta y^2} + \frac{\Phi_{j+1,k}}{\Delta x^2} - \frac{2\Phi_{j,k}}{\Delta x^2} + \frac{\Phi_{j-1,k}}{\Delta x^2} \approx -\frac{\rho_{j,k}}{\epsilon_{j,k}}$$

This linear system of equations is solved in XOOPIC by using the PETSC [PETSC] parallel matrix library, which uses MPI to communicate between processors. Similar to the electromagnetic solve, XOOPIC solves potentials on the SpatialRegionBoundaries redundantly, as in Figure 6. This redundant calculation removes the need for an additional message to communicate the results of the field solve to the adjacent region. Figure 6 also illustrates the global numbering scheme: for the purposes of the PETSc library, every mesh point in the simulation must be assigned a unique integer.

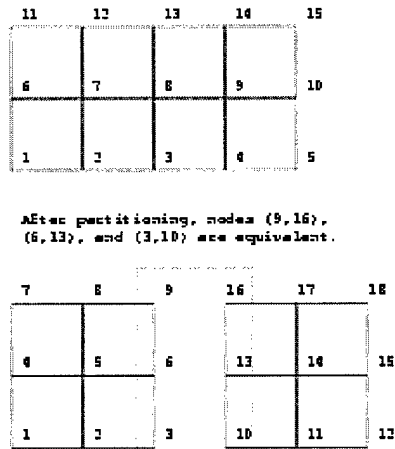


Figure 6 Indexes of the potentials on the mesh before and after partitioning. Potentials (3,10), (6,13), and (9,16) are in fact the same.

The following example equations illustrate some of the linear equations which are solved by the parallel Poisson solve. In these equations, the mesh is assumed to be uniform in both directions for simplicity.

For point #5, an interior point:

$$\Phi_4 - 2\Phi_5 + \Phi_6 + \Phi_2 - 2\Phi_5 + \Phi_8 = -\Delta x^2 \frac{\rho_5}{\epsilon_5}$$

For point #13, a point on a SpatialRegionBoundary:

$$\Phi_5 - 2\Phi_{13} + \Phi_{14} + \Phi_{10} - 2\Phi_{13} + \Phi_{16} = -\Delta x^2 \frac{\rho_{13}}{\epsilon_{13}}$$

For point #1, we will assume that the point is on a metal boundary at voltage V.

$$\Phi_1 = V$$

For point #4, a normal electric field of 0 is assumed as the boundary condition:

$$\Phi_5 - \Phi_4 + \Phi_1 - 2\Phi_4 + \Phi_7 = -\Delta x^2 \frac{\rho_4}{\epsilon_4}.$$

Slow implementation

Faster implementation

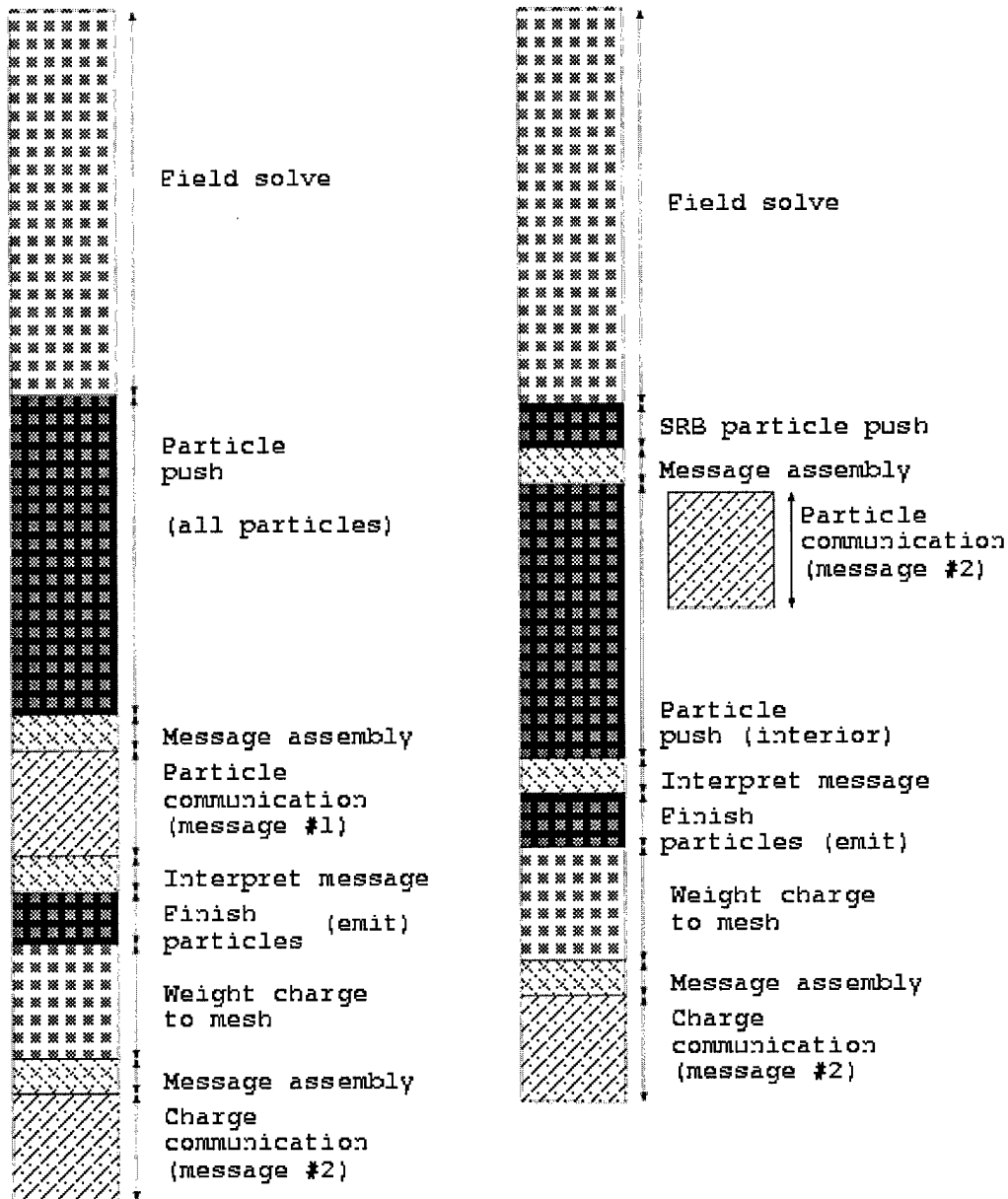


Figure 7. Diagram of computation and communication for a parallel Poisson solve.

Because Poisson's equation must be solved using global information, it is not possible to overlap field communication and the field solve in a manner similar to the "Faster Implementation" of Figure 2. Instead the entire field solve must wait until the charge density is calculated and communicated. This provides less opportunity for optimization (see Figure 7).

Furthermore, because of the global nature of the electrostatic field solve, many communications between processors must take place simply to complete the solution of Poisson's equation. This makes each electrostatic field solve take much more time than would an electromagnetic solve. It should be noted, however, that the simulation may be able to take much larger timesteps using the electrostatic field solve because of the lack of a Courant condition,

which limits the timestep of electromagnetic simulations. Figure 8 shows the flow of an electrostatic simulation in XOOPIC.

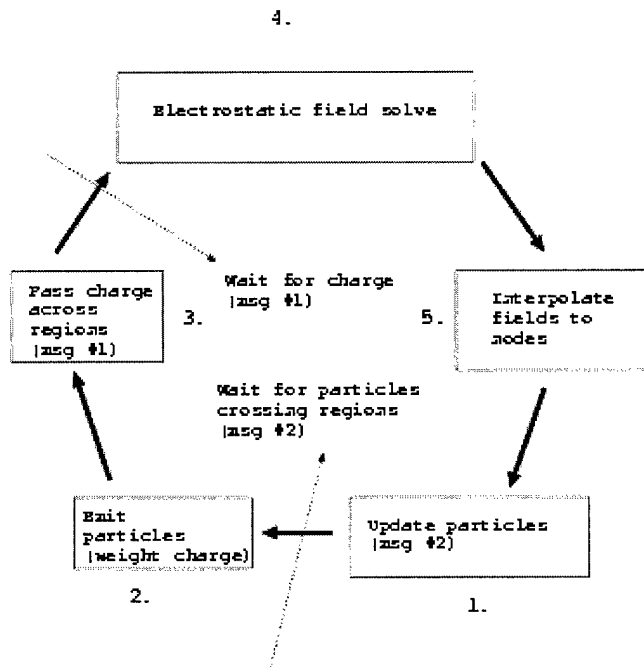


Figure 8. Flow of an electrostatic XOOPIC simulation.

#### 2.1.2.4.1 Testing of the parallel Poisson solve

#### 2.1.2.4.2 Analytic test

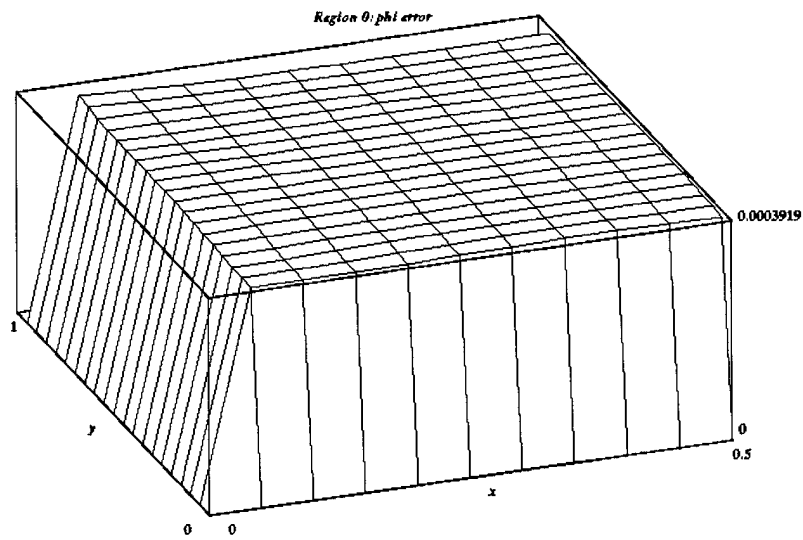
The parallel Poisson solve was first tested on the following analytic charge distribution:

$$\rho = 2\epsilon_0\pi^2 \sin(\pi x)\sin(\pi y)$$

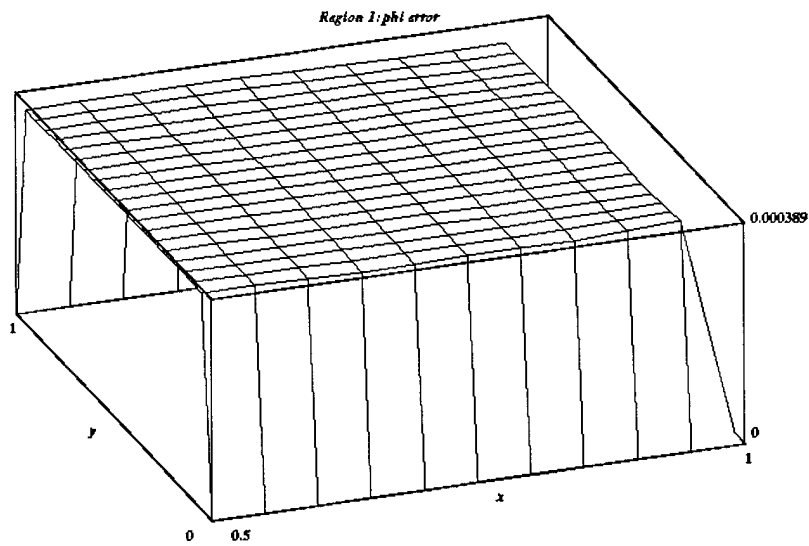
in a uniform, 1-m square box with grounded metal boundary conditions on all the walls and a relative dielectric constant of 1.0 throughout. The analytic result in this case is:

$$\Phi = \sin(\pi x)\sin(\pi y).$$

Figure 9 shows the relative error of the computations, indicating the computational result is within 0.04% of the analytical result. Figure 10 shows the absolute error. The numerical mesh used for the solution was 20x20 mesh points.



xyphi 2.20



xyphi 1.20

Figure 9. Relative error in a solution to Poisson's equation for Region 0 (left-hand partition) and Region 1 (the right-hand partition) respectively, for an analytic test case.

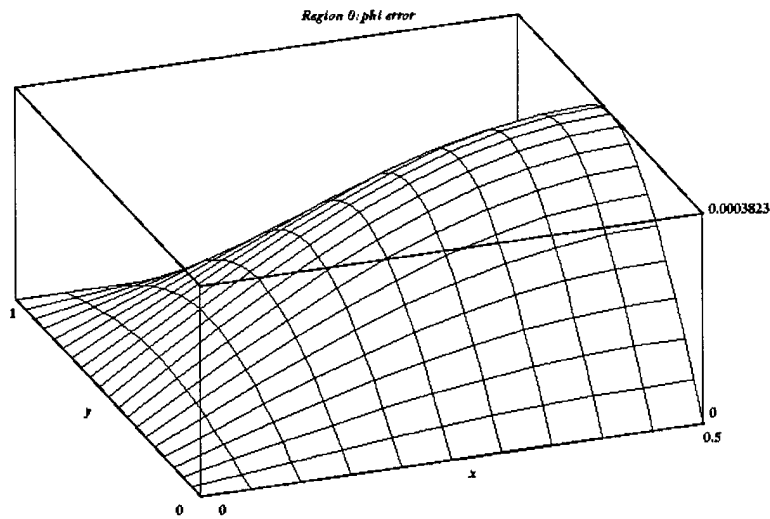
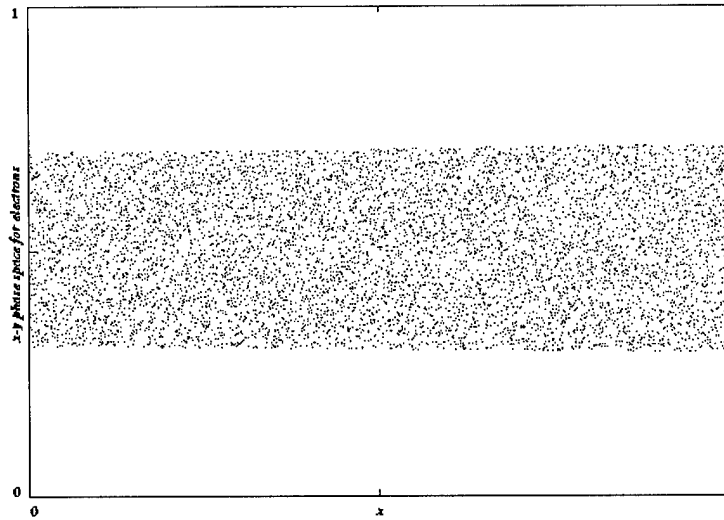


Figure 10. Absolute error in the solution to Poisson's equation in Region 0 on the analytic test case.

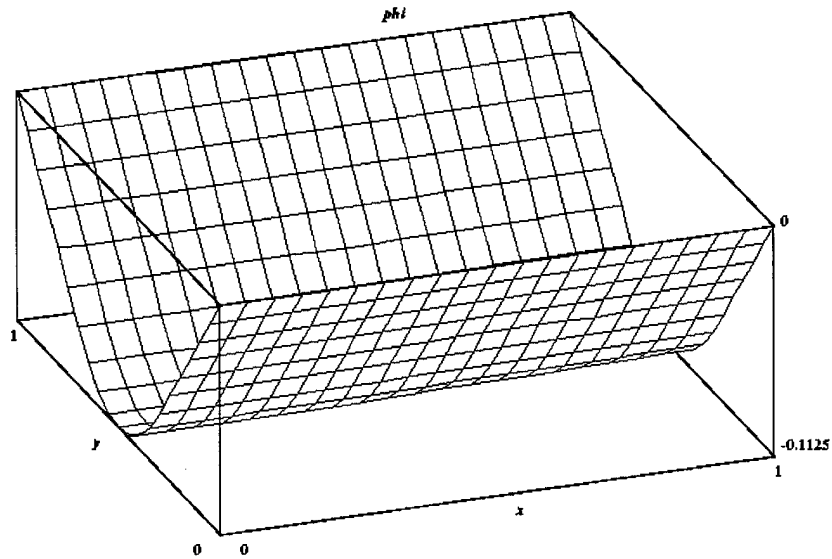
#### 2.1.2.4.3 Simulation test case.

The parallel Poisson solve was next compared with the scalar Poisson solve on an actual simulation test case. A beam is introduced on the left-hand side of the domain and propagates to the right. The top and bottom walls are grounded, conducting walls. The left- and right-hand boundary conditions are Neumann boundary conditions: which means that the electric field normal to the boundary is specified instead of the potential on these boundaries. In this case, the axial component of the electric field on the left- and right-hand boundaries is set to 0. Figure 11 shows the configuration of the simulation when the beam has propagated across. Figure 12 shows the results of the scalar Poisson solve, and Figure 13 shows the results of the parallel Poisson solve, which are in excellent (though not perfect) agreement. The reason for the difference is that a different random distribution of particles is present in the scalar vs. the parallel simulations.



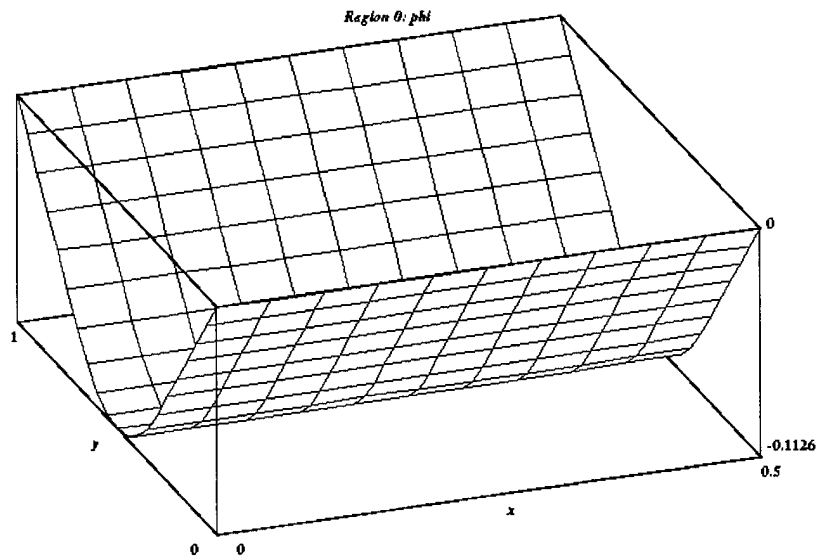
XCOPY 2.20

Figure 11. A beam of electrons has propagated across the system.

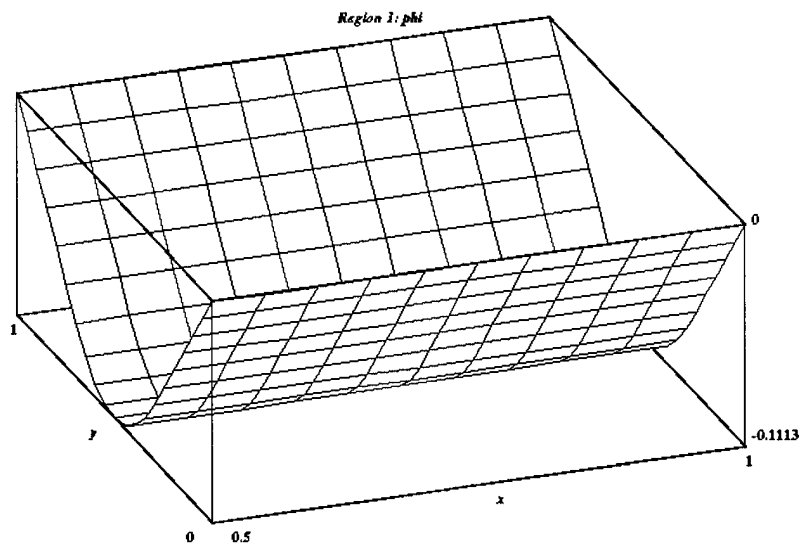


XCOPY 2.20

Figure 12. Potential in the system after a beam has propagated from left to right.



xyfig 1.20



xyfig 2.20

Figure 13. Electrostatic potential in Region 0 and Region 1 respectively. Agreement with the scalar Poisson solve is excellent.

#### 2.1.2.4 Theoretical performance limits

The ideal parallel code exhibits linear speedup with number of processors, and has single-processor performance as good as non-parallel codes. Speedup may be written this way:

$$Speedup(P) < \frac{Time(1)}{\max_i (Time(P_i) + idle_i + comm_i + extraWork_i)}$$

where "Time(1)" is the wall-clock time it would one processor to complete the task, "Time(Pi)" is the time it would take processor i to complete the work allocated to it (not including parallel overhead), "idle" is the time processor i spends doing nothing (such as when necessary data has not yet arrived), "comm" is the amount of time spent assembling and sending messages, "extraWork" is any extra work done on each processor to reduce "idle" and "comm".

#### 2.1.2.5 Load balancing

The effect of load balancing can be understood by considering two possible partitionings as in Figure 14. For simplicity, assume that updating particle positions accounts for all necessary computation. In Figure 14(a), the region is partitioned in half, but 80% of the beam is in the left half and only 20% of the beam is in the right half. In Figure 14(b), the beam is evenly partitioned.

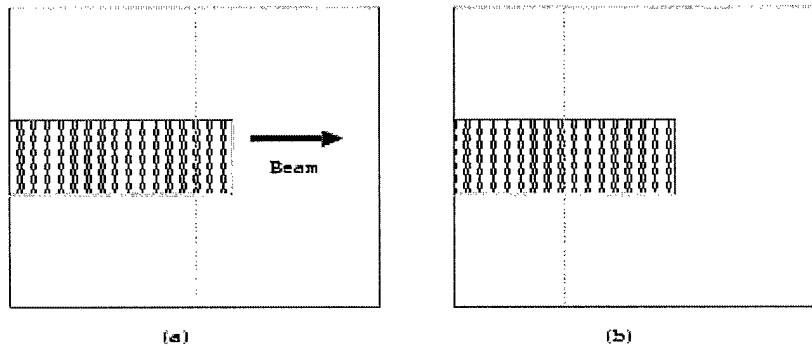


Figure 14. Unbalanced and load-balanced computational partitions.

The maximum speedup attainable in situation (a) is only 1.25, not close to 2.0 as would be hoped. As the beam propagates to the right, it is partitioning (b) which would be unbalanced: yielding only a 50% speedup when the beam has propagated all the way across.

Dynamic load balancing could be used to keep the speedup near 2 throughout the simulation: the partition can be adjusted periodically so that roughly one half of the particles are in each region. Performing this load balancing increases the overhead: it is work which never needs to be done by a non-parallel code. ICEPIC [2], a parallel code under development at AFRL, implements dynamic load balancing by moving individual cells between nodes, along with particles contained in those cells. Load balancing cannot be done this way easily in

XOOPIC, because cell quantities are stored directly in 2-d arrays: we would incur extra complexity and overhead in moving and reallocating memory. Additionally, XOOPIC does not sort particles by cell as ICEPIC does, so transferring particles belonging to a cell would require a search of all particles.

Note that in this load balancing example, where the computation of fields is assumed to be negligible, the simplest approach to achieve good load balancing is simply to avoid partitioning the region at all, and instead partition only the particles between computational nodes. However, this approach has scalability problems, and performance problems when the computation of fields is the major fraction of total computation. The scalability problem is that if the fields computation is not distributed, it must be done sequentially on one or all of the nodes, and becomes the factor limiting performance, as can be seen from the following special case of the speedup equation:

$$Speedup(P,s) < \frac{1}{s + \frac{1-s}{P}} < \frac{1}{s}$$

where “P” is the number of processors, “s” is the fraction of work which must be done sequentially, and all other considerations (communication time, overhead, etc.) are ignored.

### 2.1.2.6 Limits of scalability

#### 2.1.2.6.1 Communication time

The time it takes for a message to be communicated between nodes on a parallel computer can be modeled by the following equation:

$$CommunicationTime = L + kS$$

where “L” is a latency for a message to arrive, “k” is a constant, and “S” is the message size. L and k are strongly hardware dependent: L can range from microseconds on an SMP machine to hundreds of milliseconds on a distributed computer connected via ethernet. In the limit of many nodes and a small amount of work per node, it is L which will limit the maximum speedup attainable by XOOPIC:

$$Speedup(L) < \frac{Time(1)}{2 \times numberofsteps \times L}$$

which is completely independent of the number of nodes. Two messages must be sent per step of XOOPIC, leading to the factor of 2 in the denominator.

#### 2.1.2.6.2 Overhead due to parallelization

Parallelizing a code using message passing incurs additional computational costs which would not arise in a sequential code. These costs include CPU use to assemble messages, CPU use to package and send messages, CPU use to decode received messages, and extra, redundant work done to allow work to be done concurrently with message transmission as shown in Figure 2. These overhead costs typically scale as the message sizes:

$$Overhead = K_m + K_s M_s$$

Where  $K_m$  and  $K_s$  are machine and implementation dependent constants, and  $M_s$  is the message size.

In XOOPIE as presently implemented:

$$\text{extraWork} \approx 1.5 \times \text{EMcell} \times \text{Length}$$

where 1.5 is a constant given by XOOPIE's current implementation of SRBs, EMcell is the cost of updating the electromagnetic fields in an interior cell, and Length is the length in cells of the SpatialRegionBoundary. Therefore, for good performance on the electromagnetic field solve, the number of interior cells should be kept much larger than the number of border cells.

### 2.1.2.7 Improved PIC loop, similar to one used in ICEPIC

The PIC loop shown in Figure 4 can be improved to allow for load balancing between fields and particles, as is done in ICEPIC, but not XOOPIE. Consider a case as in Figure 15 where one computational region has many particles (Region 1), but is small, so its field calculation is negligible, and a large computational region, with few particles, so the particle calculation is negligible (Region 2).

In XOOPIE, because the fields update cannot complete until all the particles are moved and J is computed, computation for Region 2 would have to idle until the Region 1 particle push completed.

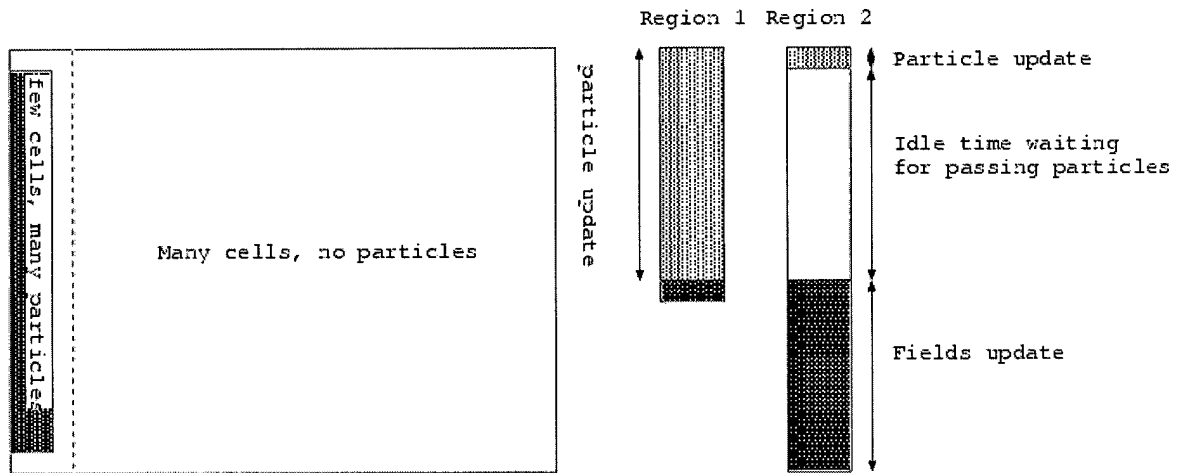


Figure 15. Case where fields and particles are very unevenly distributed between computational regions.

The PIC loop shown in Figure 16 allows load balancing between fields and particles: a large interior field solve could be performed on one node concurrently with a large particle update on another, reducing to one the number of synchronization points in the loop.

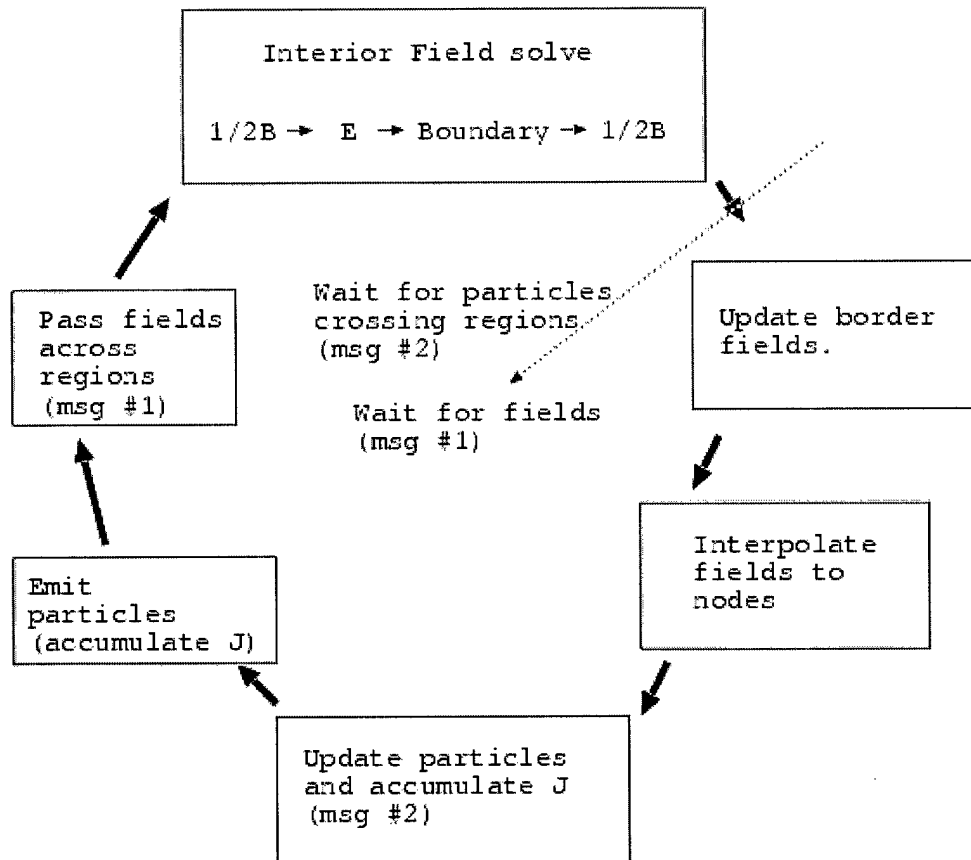


Figure 16. Improved PIC loop, allowing balancing between fields and particles.

### 2.1.2.8 Status of parallel XOOPIC

Parallel XOOPIC is presently available to the general public at <http://ptsg.eecs.berkeley.edu/xoopic/xoopic.html>. The parallel version has these additional features over the non-parallel version:

- Partitioning in either x or y direction. Partitioning in both simultaneously is implemented but not tested.
- Parallel electromagnetic field solve.
- Parallel electrostatic solve in x-y geometry (r-z will likely operate properly when a problem in the PETSc library is fixed).
- Parallel particle push.
- Particle passing across virtual boundaries.
- Automatic partitioning of a given model.
- Diagnostics by computational region.
- Simulation checkpointing.

We tested parallel XOOPIC on a well-load-balanced case on several SMP machines, and on a single-processor DEC Alpha. We observed near-linear speedup or even super-linear speed up (Figure 17). In particular, 2 CPUs on a 4-CPU Pentium Pro machine performed more than twice as well as 1 CPU. This may be due to the larger amount of cache available. The two 8-processor runs exhibit the difference between the GNU g++ compiler and the proprietary Sun compiler on an Ultra Enterprise 5000. 942,000 particles were used in all tests, so the 8-CPU tests had roughly 120,000 particles per CPU.

### Particle Pushes per Second vs. #CPUs

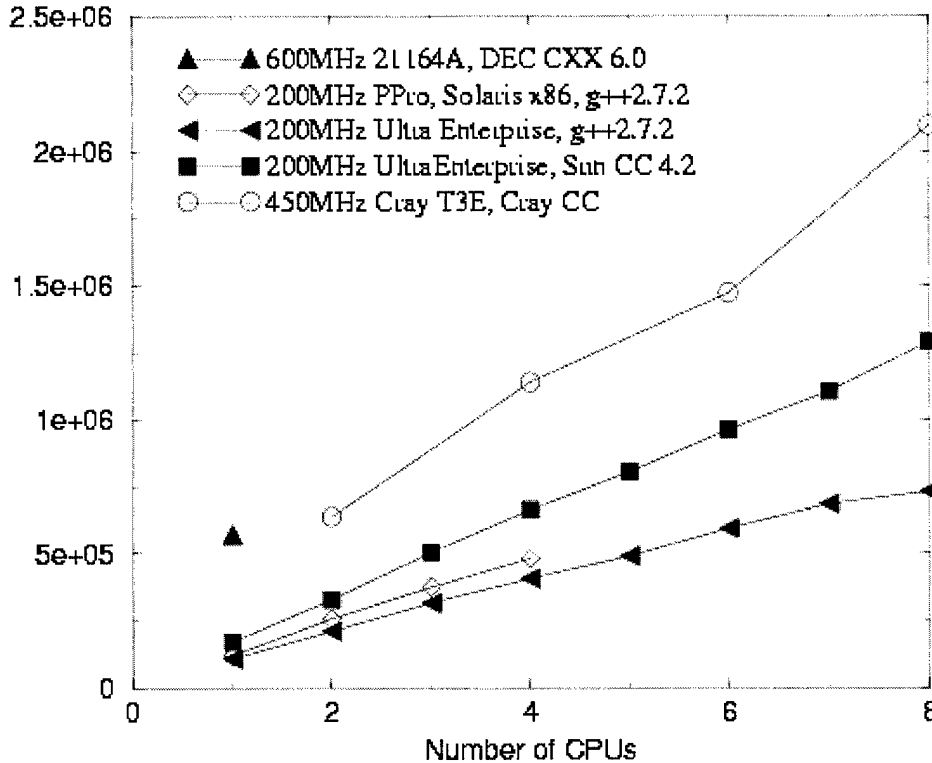


Figure 17. Scalability of parallel XOOPIC.

The parallel version (as of XOOPIC 2.52) has these shortcomings:

- Dielectric triangle objects are not split correctly across regions.
- Beam emitter boundaries cannot be split by SRBs.
- Particles see nearest-grid-point B3 near SRBs rather than linearly-weighted B3.

#### 2.1.2.9 Future Work

Efforts to extend parallel XOOPIC are ongoing. Areas being worked on include:

- Fixing identified bugs.
- Improved diagnostics.

- A parallel r-z Poisson solve, so that r-z electrostatic models may be treated.
- Dynamic load balancing for best use of parallel resources.
- Elimination of the need to wait for the particle update to complete before identifying which particles must be communicated.

Implementation of the parallel Poisson solve is under way, using the PETSc library with BlockSolve95 has been completed for the x-y and r-z models, but a bug in the PETSc library causes failure on the cylindrical model.

Elimination of the need for the particle update to complete and dynamic load balancing are performance optimizations which will extend the usefulness of parallel XOOPIIC. Presently, scalability to many CPUs is limited: all computation must wait for the particles to be transferred between SRB pairs, and no dynamic load balancing is done. Efficient use of parallel resources thus requires some planning on the part of the user at present: the type of auto partitioning used can ensure good load balancing, and for some models, reasonable load balancing may not be possible. Dynamic load balancing will improve matters by distributing workload at runtime.

#### **2.1.2.10 Acknowledgments**

This work has been funded by AFOSR F49620-94-1-0387, and AFOSR/MURI F49620-95-1-0253. We are also grateful to Jim Havranek (formerly at AFRL) and Gerry Sasser (AFRL) for their useful comments and suggestions.

#### **2.1.2.11 Bibliography**

J.P. Verboncoeur, A.B. Langdon, and N.T. Gladd, "An object-oriented electromagnetic PIC code", *Computer Physics Communications* 87 (1995) 199-211.

G.E. Sasser III, J.J. Havranek "Load-balanced single domain decomposition parallel electromagnetic PIC", Private Communication, May 1999.

"The Message Passing Interface (MPI) Standard", <http://www.mcs.anl.gov/mpi/>

[PETSC] Satish Balay and William D. Gropp and Lois Curfman McInnes and Barry F. Smith, {PETSc} home page, "<http://www.mcs.anl.gov/petsc>", 2000.

"The PETSc Library", <http://acts.nersc.gov/petsc/main.html>

"BlockSolve95", <http://www.mcs.anl.gov/mpi/>

#### **2.1.3 Three-Dimensional Model**

This section describes the project to extend the existing XOOPIIC code from two to three spatial dimensions. The extension of the XOOPIIC code to three dimensions provides a tool for modeling many devices which are presently beyond the scope of existing tools. The coordinate systems of interest here include Cartesian (x-y-z) and cylindrical (r-theta-z). Such a project would be beneficial to the HEM effort since it would enable modeling of novel concepts such as multi-beam and other inherently three-dimensional devices.

This project entails the extension of XOOPIIC from 2d to 3d, including the flexible internal structures and existing models. In 3d, the code will be capable of modeling non-axisymmetric devices, higher order modes in axisymmetric devices, and planar sheet-beam devices which are bounded in the third dimension such as the proposed Stanford University klystrino.

The fully kinetic code will include electrostatic and electromagnetic field models, appropriate for gun, circuit, and collector modeling. Devices which require 3d are numerous, including many gyro-devices, sheet-beam devices, multi-beam/multi-circuit devices, and devices with realistic (3d) inputs and outputs. The development effort can proceed asynchronously with the parallel XOOPIC extension effort, but will greatly benefit from its performance enhancement on significant computational problems. This project is most effectively performed in conjunction with the modeling of a specific 3d device (proposed below), which serves to motivate the work. A 3d device with theoretical and especially experimental support can also be invaluable in benchmarking the development.

The enclosed paper, P. J. Mardahl, J. P. Verboncoeur, and C. K. Birdsall, "Progress on a 3d particle-in-cell model of a W-band klystron", provides complete details of the three dimensional code design.

In particular, the field locations are defined on the Yee mesh, and the solution of the Poisson's equation as well as Maxwell's equations are described for the XOOPIC code. In addition, the integration of the particle equations of motion are described for fully relativistic particles. The coupling of the particles to the fields is described, including alternative techniques for improving the noise properties of the coupling.

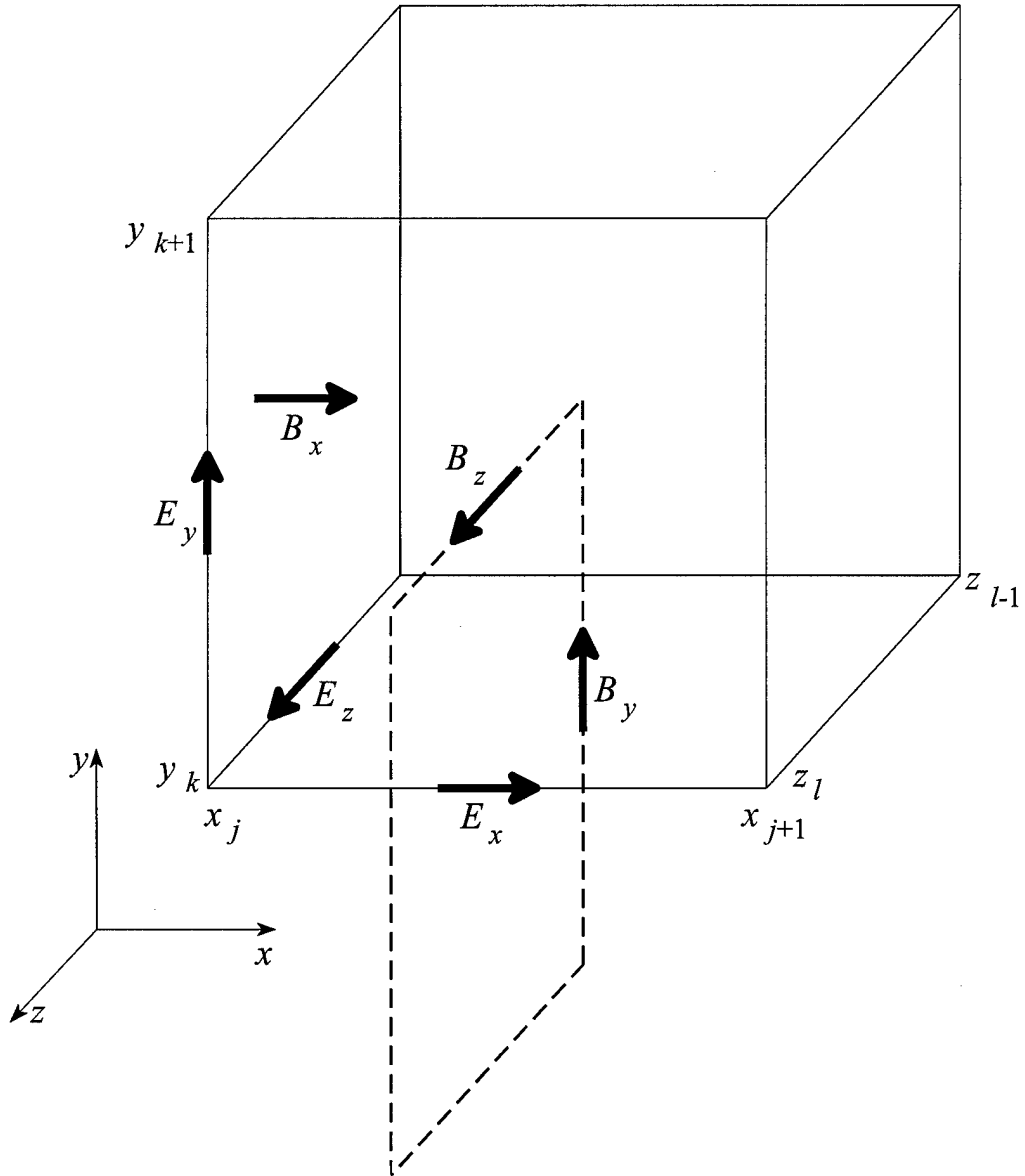


Figure 18 Yee mesh depicting XOOPIC field locations in three dimensions.

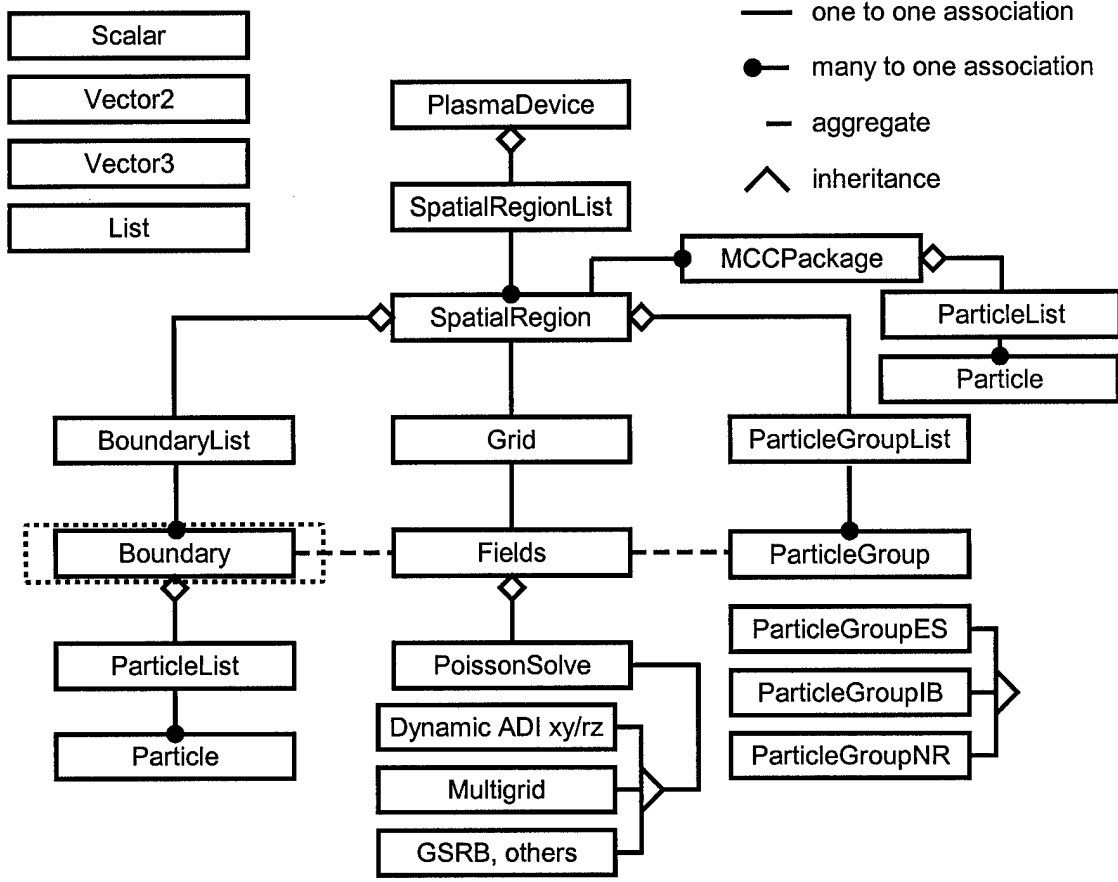


Figure 19 XOOPIC code structure.

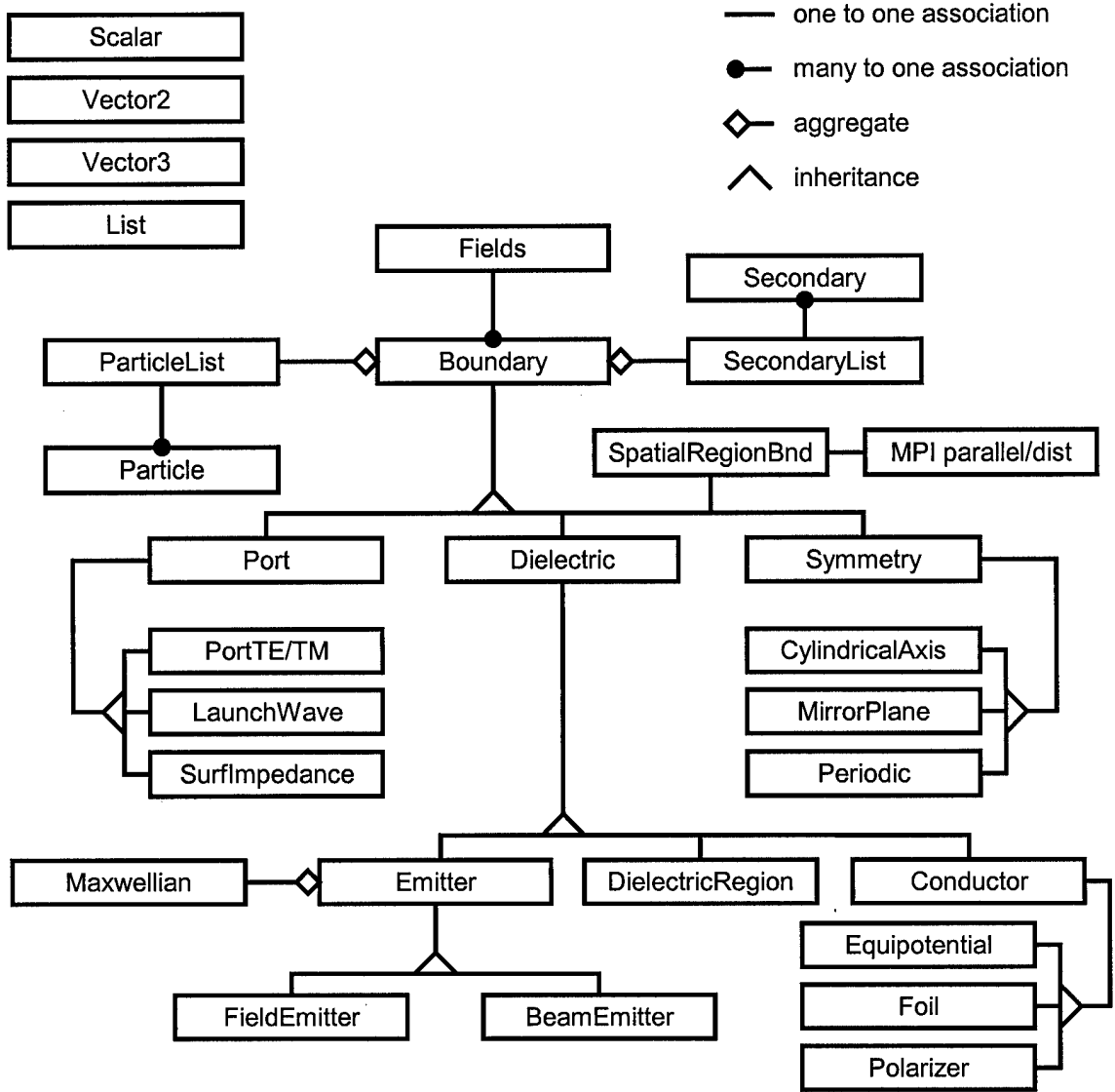


Figure 20 Detail of XOOPIC boundary condition code structure.

Currently, the field equations have been implemented in Cartesian coordinates, as well as cylindrical coordinates. The field-related boundary conditions are presently limited to ideal conductors. A surface impedance boundary condition will be completed for a paper to be presented at IEEE ICOPS in June, 2000. This boundary condition provides the capability to launch waves from a specified impedance, as well as a lossy surface with a specified impedance for absorption of electromagnetic energy.

The particle equations of motion have been implemented in both Cartesian and cylindrical coordinates. Particle boundary conditions include emitters for generating beams of various profiles including time-dependent current capability. Other boundary conditions include ideal absorption of particles at boundaries. Secondary emission is not yet functional.

The current implementation allows specification of the model from the input file using the namelist format implemented in the 2d version. The primary modification for 3d is in specifying the physical dimensions of the system. The present model describes boundaries and system edges

as a series of intersecting planes orthogonal to the coordinate system. A superior model, such as specification of common three-dimensional objects such as cubes and annular objects is under consideration for a future implementation.

The integration of the above components will be demonstrated in the forthcoming Mardahl IEEE ICOPS paper. This provides sufficient capability for device-level algorithm verification, including verification of the electromagnetic solver, particle algorithms, boundary conditions, and a device level model. The device level model is the Stanford University klystrino described under below.

A number of issues must be addressed in future work. As the number of dimensions increases, issues like noise reduction become increasingly important. In higher dimensions, it becomes more challenging to model sufficient particle statistics to achieve an acceptable noise level. This makes techniques like filtering of particle source terms ( $\rho$  and  $J$ ) as well as higher order weighting schemes, crucial for reducing the effects of particle statistics. Extension of the 2d 1-2- filtering technique is planned for 3d, as well as higher order weighting schemes and extension of the 2d temporal filtering noise reduction model. Furthermore, the number of particles and the number of grid cells result in memory challenges, which require massively parallel computing capability to solve larger problems (see the Parallel PIC section for details).

#### **2.1.4 Current Injection Algorithms (25% MURI funded)**

This project was funded in the early years of the MURI program, and funding was switched to another AFOSR-related contract. The objective of the project was to achieve full second order accuracy for particle injection. Existing algorithms were first order accurate in time or space, and in some cases zero-order errors were present.

The problem was first measured in a crossed-field diode model, where the virtual cathode stability was critically dependent upon the injection algorithm. Further investigation revealed that the same problem was present in all injection models, including magnetized and unmagnetized, space charge dominated or not.

A number of schemes were analyzed and the algorithms were developed to inject particles to achieve proper time centering and second order accuracy in both space and time, consistent with the leap frog algorithm used to advance particles after the injection timestep. Furthermore, the accuracy constraints were also imposed for the partial timestep injection required to obtain temporally uniform current injection. The models described can be applied to both electrostatic and electromagnetic models.

Since the details of these algorithms have been described in detail in a paper accepted for publication in *J. Comput. Physics*, we do not describe the details herein but rather attach the preprint: K. L. Cartwright, J. P. Verboncoeur, and C. K. Birdsall, "Loading and Injection of Maxwellian Distributions in Particle Simulations", accepted for publication in *J. Comput. Physics* (2000).

#### **2.1.5 Moving Window Algorithm (1% MURI funded)**

The moving window algorithm enables XOOPIC to simulate a moving system in a moving frame of reference. This is useful for following beams traveling over long distances, interacting with periodic structures, and observing the growth of phenomena over long distances which would normally be prohibitive to simulate. This project leverages many of the technologies developed under the HEM MURI. The moving window project may be beneficial to the MURI

objectives, although it was primarily funded by outside sources. It is included here for informational purposes.

Some devices are too large to be simulated efficiently using the PIC method: a full simulation of the entire device may be impractical due to limits on memory and CPU speed, even given a large parallel computer. An alternative is to follow a group of particles as it traverses the device, and simulate only those particles and the nearby portions of the device, as in Figure 21. This simulated subset of the entire device is the moving window.

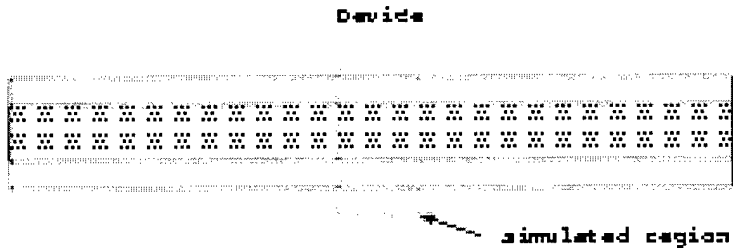


Figure 21. The moving window follows a subset of particles through the device.

There are several approaches to implementation of a moving window. One is to move the mathematical mesh along with the particles, and give the background and walls a velocity relative to the mesh. Another is to keep the mesh stationary with respect to the background and create new mesh on the leading edge and discard it on the trailing edge, while the particles of interest are still moving quickly with respect to the mesh.

In the former approach, major modifications to the field solve and the particle push would be necessary in order to implement a moving window in XOOPIC. The latter approach is far more straightforward, and is the method used in XOOPIC. No modifications to the basic field solve and particle push are necessary; however, additional functionality must be added.

Instead of actually creating and destroying mesh in XOOPIC, which would be an expensive operation because memory would have to be allocated and deallocated, field values are shifted from "upstream" mesh points to "downstream" mesh points. I.e., for a moving window which is following a group of particles moving to the right, analytic new fields are introduced into the rightmost mesh point, and the fields in the rightmost mesh point are copied to the one immediately left of it, etc. Copying actually begins at the leftmost mesh point and proceeds to the right, as is shown in Figure 22.

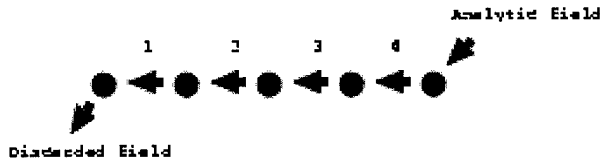


Figure 22. For a rightward-moving window, fields are copied in the direction indicated by the arrows, starting with copy #1 and proceeding to the right.

When this shift in the fields takes place, all the particles must also be shifted, or else they would see an instantaneous, unphysical change in the fields near to them. At this time, any particles in the leftmost cell (for a rightward-moving window) are discarded, for they have left the moving window. New particles may be introduced in the rightmost cell, in order to represent a background plasma, if desired.

### 2.1.5.1 Boundary conditions

Boundary conditions present the major difficulty with this moving window approach. However, the difficulty is removed if the moving window is moving at or close to the speed of light, a common case of interest in situations where use of a moving window is desirable.

In the case of a rightward-moving window moving at the speed of light, disturbances at the leftmost boundary cannot propagate into the moving window domain faster than they are discarded, because all electromagnetic waves are constrained to move with a velocity less than or equal to the speed of light.

Similarly, incoming fields on the right hand side are not affected by the contents of the moving window to the left of it: fields here may be safely specified analytically in a simple way. XOOPIC handles the boundary conditions in this manner.

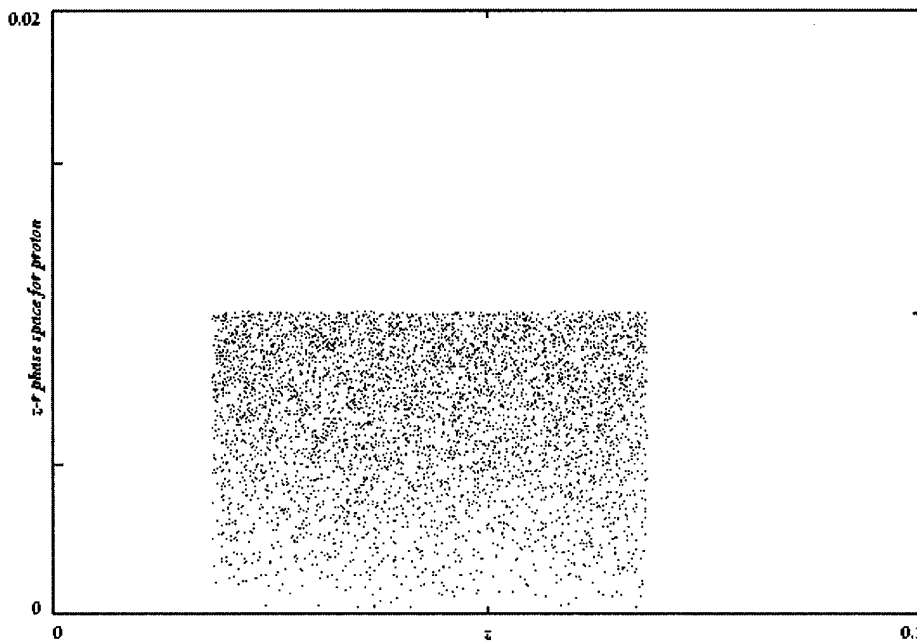


Figure 22. This group of protons moving at nearly the speed of light has been kept in the moving window for ten transit times.

### 2.1.5.2 Moving window and parallel XOOPIC

Combining parallel operation and the moving window leads to some additional complication. Whenever a shift in fields takes place, (usually every timestep, or every few timesteps), the shifted fields and particles must be passed to the downstream computational region. This requires extra communication between processors and therefore reduces the maximum achievable parallel efficiency. Figure 23 shows the modified PIC loop with the shift communication added. At present, the shift communication halts computation until it has completed instead of allowing concurrent computation on local data. Figure 24 illustrates the extra cost incurred by the messages carrying the shifted data.

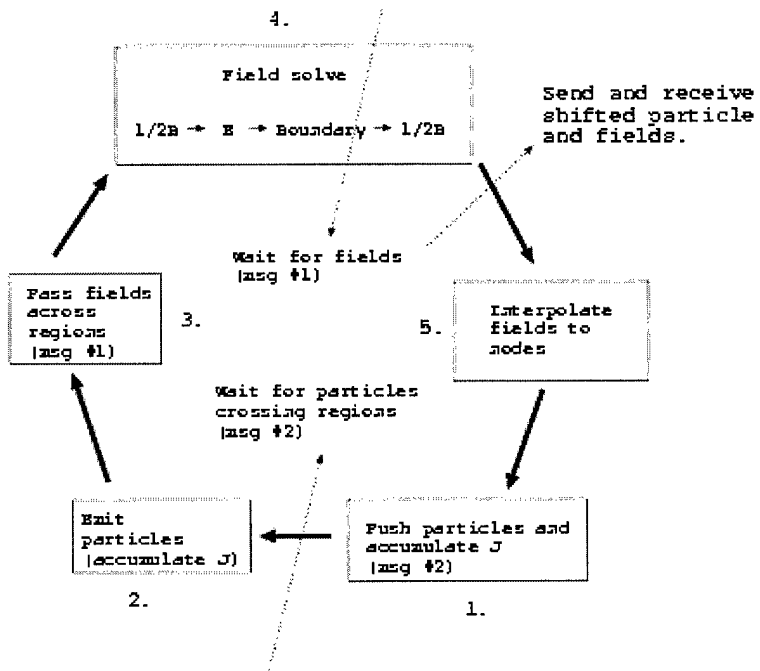


Figure 23. Modified parallel PIC loop with messages for fields and particles shifting.

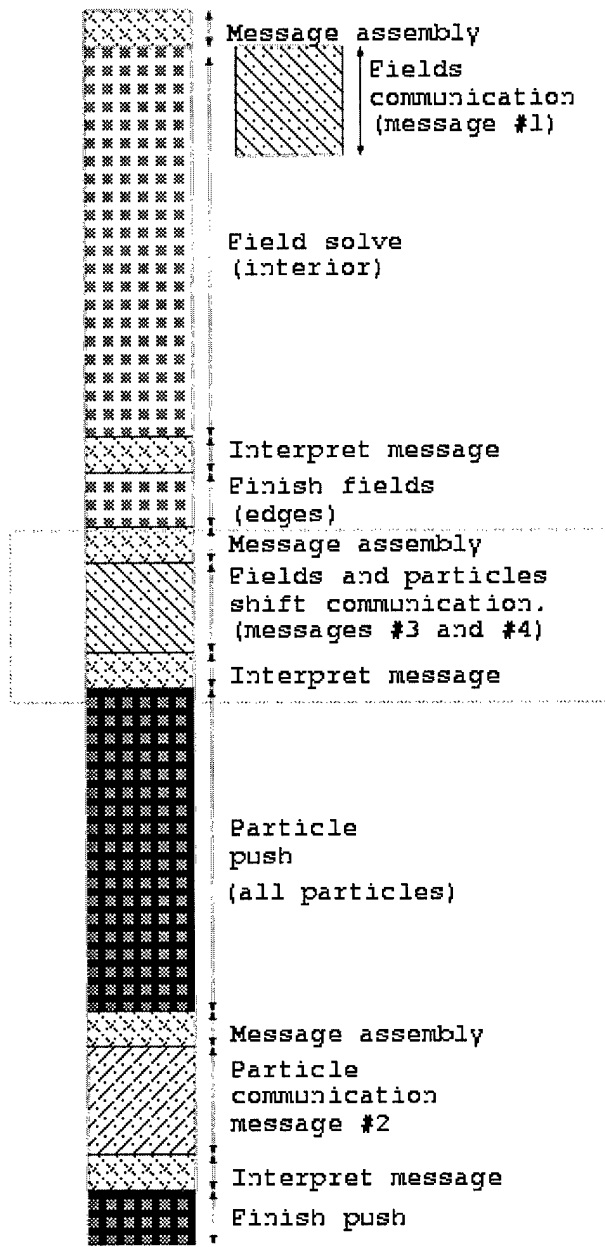


Figure 24. Diagram of messages and computation in XOOPIC parallel PIC with the moving window.

## 2.2 Multipactor Modeling

This section describes the progress of the multipactor research at University of California at Berkeley. The secondary emission model has been described in previous reports, as well as the upcoming section in a book: J. P. Verboncoeur, "Secondary Emission Model", Chapter 11: Modeling and Computational Techniques, in *Advances in High Power Microwave Sources and Technologies*, ed. R. J. Barker and E. Schamiloglu, IEEE Press.

The chief element of progress in this reporting period has been the simulation of multipactor phenomena on single surface dielectric media, such as microwave windows. The new and unique element of research was the inclusion of self-consistent space charge effects for the first time.

Multipactor is relevant to the HEM effort since multipactor can limit the performance of microwave beam devices. Multipactor is a phenomenon which occurs in moderate rf fields, either between two surfaces (metals) or at a single surface (dielectrics, such as windows). Both single and two-surface multipactor phenomena are studied. In HEM devices, the field strength in gaps is usually too large to allow multipactor; however, some distance from gaps appropriate field strengths can occur. In addition, the secondary model developed for the multipactor model can be applied to other secondary emission phenomena such as beam interception.

### **2.2.1 Single-Surface Multipactor (Dielectrics)**

The single surface multipactor occurs on dielectric surfaces, such as microwave windows<sup>2,3,4</sup>. An rf electric field transverse to the surface provides the energy source, while a dc electric field normal to the surface occurs due to charge separation of the emitted secondary electrons and the net-positively charged dielectric surface.

An electron incident on the surface emits a secondary electron with some finite energy. The dc field normal to the surface is generated between to the negative space charge and the positively charge dielectric surface. The dc field decelerates the electrons in the normal direction, while the rf field accelerates the electrons in the transverse direction. As the dc field finally draws the electrons back to the dielectric surface, the electrons in the appropriate phase have gained sufficient energy to generate additional secondaries.

Space charge is the dominant saturation mechanism for the multipactor, rather than beam loading of the rf fields. The self-consistent effects of beam loading and the finite transverse dimension of the dielectric surface are not included in previous works, nor are the full self consistent effects of the space charge on timescales fast compared to the rf period.

In this work, we have extended the existing models to the fully self consistent regime using the XOOPIC PIC code in two dimensions and the XPDP1 code in one dimension. The work was performed in collaboration with Lau et al. at the University of Michigan. Recently, we have verified the experimental measurements of Texas Tech. Researchers, which indicated that even at modest power levels x-rays were observed. This is a result of space charge shielding of the dc field, resulting in long flight times between impacts. We were unable to corroborate the Texas Tech. calculation that indicated a small change in applied field resulted in a large change in impact energy profile. However, we discovered an upper energy cutoff, supported by a simple theory.

The details of the research are presented in a paper accepted for publication: A. Valfells, J. P. Verboncoeur, and Y.Y. Lau, "Space charge effects on multipactor on dielectric", accepted for publication in the IEEE Transactions on Plasma Science 8<sup>th</sup> Special Issue on the High Power Microwave Generation (2000).

## **2.3 Modeling Field Enhancement in an RF Gap**

### **2.3.1 Introduction**

The high electric fields applied to microwave cavities induce field emission of electrons. This field emission current  $I_{FE}$  combined with neutral desorption at the nose cones of a

microwave cavity can lead to surface damage. The field emission current heats the metal surfaces, leading to desorption of neutral contaminants on the surfaces. The field emitted electrons ionize the desorbing neutrals. The resulting positive ions enhance the field at the emitter, increasing  $I_{FE}$ . The higher  $I_{FE}$  increases the power dissipation and the temperature of the emitter, leading to more neutral desorption. More neutrals imply more positive ion creation and field enhancement, etc., leading to positive feedback loop. Eventually, the emitter surface will melt and is "rf processed".

We use a 1d3v particle-in-cell/Monte-Carlo collisions (PIC/MCC) model to show the effect of positive space charge on  $I_{FE}$ . The model consists of two parallel plates; the left plate is driven by a sinusoidal rf voltage source and the right plate is grounded.  $I_{FE}$  is given by the Fowler-Nordheim relation. Our first model assumes a constant and uniform neutral background while a second model takes neutral flows and gradients into effect. In both models, we observed field enhancement due to positive ion formation. We assume that hydrogen atoms desorb from the copper surfaces of the cavity. The 1d3v PIC/MCC simulation code PDP1 self-consistently solves for the fields due to the applied rf and the charges [1].

### 2.3.2 Field emission

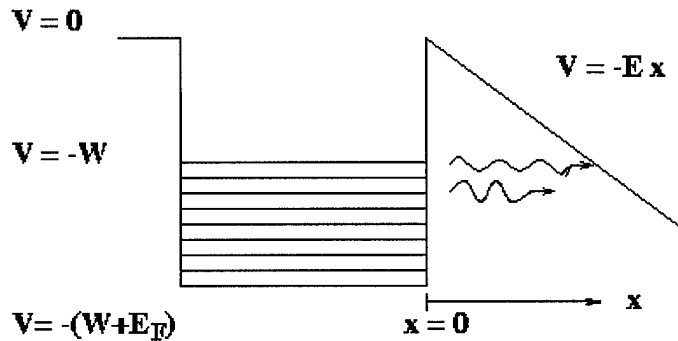


Figure 25: Electrons may tunnel through the potential barrier, especially where the barrier is thinner.

Electrons may be extracted from conductors by applying a strong electric field. Applying a high field to the metal produces a triangular potential barrier through which electrons at the metal surface may tunnel quantum mechanically, especially where the tunnel is thinner. (See Fig. 25). By solving Schrodinger's equation, Fowler and Nordheim obtained the barrier penetration probability  $D(\epsilon)$ , where  $\epsilon$  is the kinetic energy of the electrons in the metal. By multiplying  $D(\epsilon)$  by the number of electrons arriving at the surface with kinetic energy  $\epsilon$  and integrating over all  $\epsilon$ , Fowler and Nordheim derived the "Fowler-Nordheim" equation relating field emission current density  $j_{FE}$  ( $A/m^2$ ) with emitter electric field  $E$  (V/m) and work function  $W$  (eV) [2].

$$j_{FE} = \frac{1.54 \times 10^{-6} \times 10^{4.52W^{-1/2}} E^2}{W} \exp\left(-\frac{6.53 \times 10^9 W^{3/2}}{E}\right), \quad (1)$$

Typically, field emission occurs at fields of the order of  $10^9 - 10^{10}$  V/m.

Fowler and Nordheim calculated the current for a cold flat surface. The current is weakly dependent on temperature, but it is strongly dependent on emitter shape. To take shape into account, there is a geometric field enhancement parameter  $\beta = E/E_{applied}$ , the ratio of the local emitter field over the applied field. Plugging this into Eq. (1), we obtain,

$$j_{FE} = \frac{1.54 \times 10^{-6} \times 10^{4.52W^{-1/2}} (\beta E_{applied})^2}{W \cdot 36} \exp\left(-\frac{6.53 \times 10^9 W^{3/2}}{\beta E_{applied}}\right), \quad (2)$$

### 2.3.3 Field enhancement

The following flow chart (Fig. 26) illustrates the positive feedback loop that may lead to “rf processing” of the emitter. The field emission of electrons heats the metal emitter, leading to desorption of neutral contaminants. The electrons collide with the neutrals and create positively charged ions near the emitter. The ions neutralize the self-field of the emitted electrons and also enhance the electric field, creating larger field emission currents  $I_{FE}$ . Ions also provide a dc bias so that the fraction of the rf cycle during which field emission is active increases, leading to larger average field emission currents. Power dissipation at the emitter will increase with increasing emission current. This will increase the temperature of the emitting surface, leading to more neutral desorption. This increase in neutral flux will increase the ionization rate which will increase the emission current, and so on, causing a positive feedback loop. At some point the emitter temperature will reach its melting point and the emitter surface is “rf processed” [3].

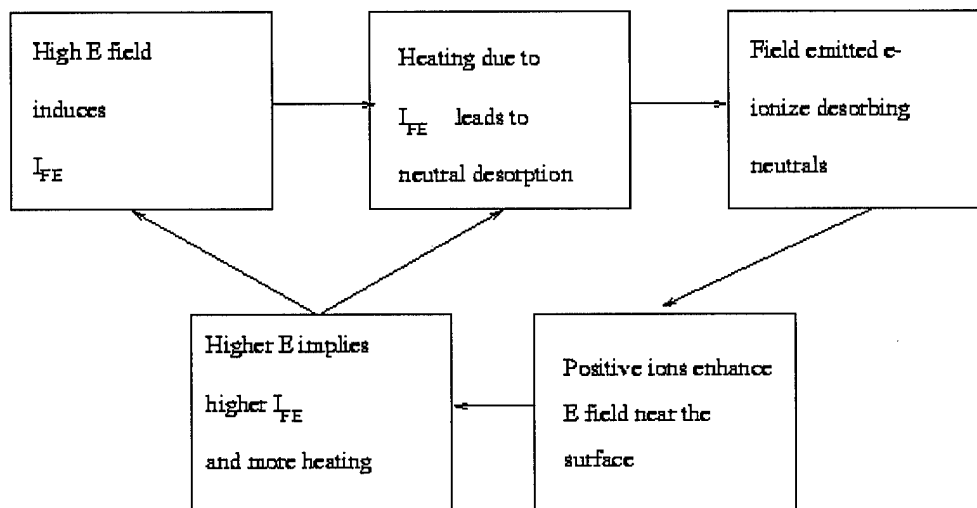


Figure 26: Positive feedback loop leading to “rf processing” or melting of the emitter surface.

### 2.3.4 RF gap simulation model

We modeled the nose cones of the microwave cavity with two parallel plates of diameter 5 mm and spacing of 2 mm. The initial applied field was 120 MV/m and the applied frequency was 11.424 GHz. We assumed that the gap was filled with atomic hydrogen. Though pressure within the entire cavity is low ( $\sim 10^{-8}$  Torr), the neutral pressure may be high near the surface of the emitter when neutrals desorb from the surface.

We are mainly interested in the region near the field emitter. At 120 MV/m, the electron energy is already 1200 eV after 10  $\mu\text{m}$ . The electron impact ionization cross section peaks at about 100 eV and then starts to decline. Also, we get significant neutral density only near the emitter where the neutral desorption occurs. Thus, most of the ionizations occur within a few  $\mu\text{m}$  of the emitter. Thus, we limited our simulation to a 10  $\mu\text{m}$  region near the emitter.

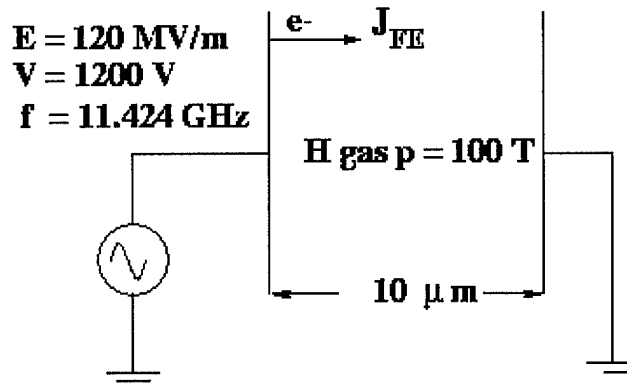


Figure 27: RF gap simulation model near emitter region.

Since the electrons reach maximum energies of about 1200 eV, corresponding to velocities of about 0.05  $c$ , we neglected relativistic effects and the self magnetic field of the electrons, and used an electrostatic field solve. To avoid having electrons move more than one cell per timestep, and to resolve submicron distances, we used femtosecond ( $10^{-15}$  sec.) timesteps.

The Fowler-Nordheim current density is given by,

$$j_{FE} = \begin{cases} AE_{applied}^2 \exp(-B/|E_{applied}|), & \text{for } E_{applied} < 0 \\ 0, & \text{for } E_{applied} > 0 \end{cases} \quad (3)$$

where  $E_{applied}$  is the instantaneous electric field at the emitter site, and  $A$  and  $B$  are input parameters supplied by the user and which depend on the work function  $W$  and geometric enhancement factor  $\beta$  of the emitter. We assumed that the metal was copper with  $W = 4.59$  eV. We also assumed a  $\beta$  of 50. (See Fig. 3 of Reference [2]). The electrons emitted from the surface were assumed to have a temperature of 500 C. Furthermore, the electron drift velocity normal to the surface was randomly chosen from the range 0.1 V to 1V.

Our collision model was the standard Monte Carlo collisions (MCC) package [1]. We included only electron-hydrogen atom collisions for ionization, excitation and elastic scattering. This may be improved by allowing more types of collisions in the future. However, the main reaction of interest is the ion production from ionization.

We also assume that enough neutrals have desorbed in the region near the emitter surface to produce a high neutral pressure of about 100 Torr. As a first approximation, we assume a constant uniform neutral pressure in this 10  $\mu\text{m}$  region. A monolayer of hydrogen atoms has a surface density of about  $1.5 \times 10^{19}$  atoms/ $\text{m}^2$ . (See for example Table 2.17 of reference [4].)

A suddenly released monolayer will form a dense expanding gas in the 10  $\mu\text{m}$  region of approximate density,  $n_g = 1.5 \times 10^{19}$  atoms/ $\text{m}^2 \div 10 \mu\text{m} = 1.5 \times 10^{24}$   $\text{m}^{-3}$ . The emitter surface has an initial temperature of about  $T=500$  C= $737$  K. Thus, the pressure in the cavity is approximately:  $p = n_g k_B T \approx 1.5 \times 10^{24} \text{ m}^{-3} \times 1.38 \times 10^{-23} \text{ J/K} \times 737 \text{ K} \approx 1.5 \times 10^4 \text{ Pa} \approx 100 \text{ Torr}$ .

### 2.3.5 Simulations with constant and uniform neutral background

We used the rf gap model described above to do some PIC/MCC simulations. The simulations were conducted for several rf cycles. We considered two cases.

- Case 1: Collisions are turned off so that there are no ions created in the gap. There is no field enhancement at the emitter.
- Case 2: Collisions are turned on so that ions are created near the emitter. The positive ions enhance the field at the emitter. The ions also produce a positive dc bias so that the fraction of the rf cycle in which the field emission is turned on ( $E < 0$ ) increases. (See Fig. 28).

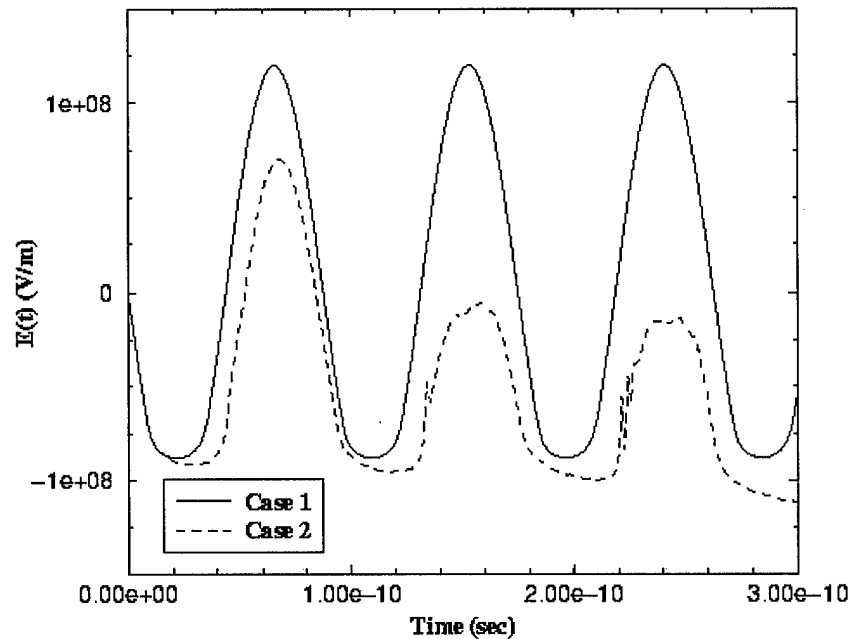


Figure 28: Electric field at emitter vs. time for cases 1 and 2. A constant and uniform neutral background is assumed for this set of simulations.

### 2.3.6 Simulations with neutral flow and gradient

We improved our PIC/MCC model by incorporating a time varying neutral flow. At  $t=0$ , a monolayer of  $1.5 \times 10^{19}$  H atoms/m<sup>3</sup> is released from the emitter. The emitted neutrals are assumed to have a Maxwellian velocity distribution with temperature  $T_g = 500 \text{ C} = 737\text{K}$ . As in the previous section, we conducted two cases. In case 1, the collisions are turned off so that no ions are created in the gap. In case 2, the collisions are turned off so that ions are created near the emitter. As with the previous simulations, the ions enhance the field at the emitter and generate a dc bias so that the fraction of the rf cycle during which field emission is turned on increases. However, it takes a couple of rf cycles for the field enhancement to begin because the neutral atoms need time to flow into regions in which the electrons have reached ionization energies ( $> 13.6 \text{ eV}$ ). (See Fig. 29).

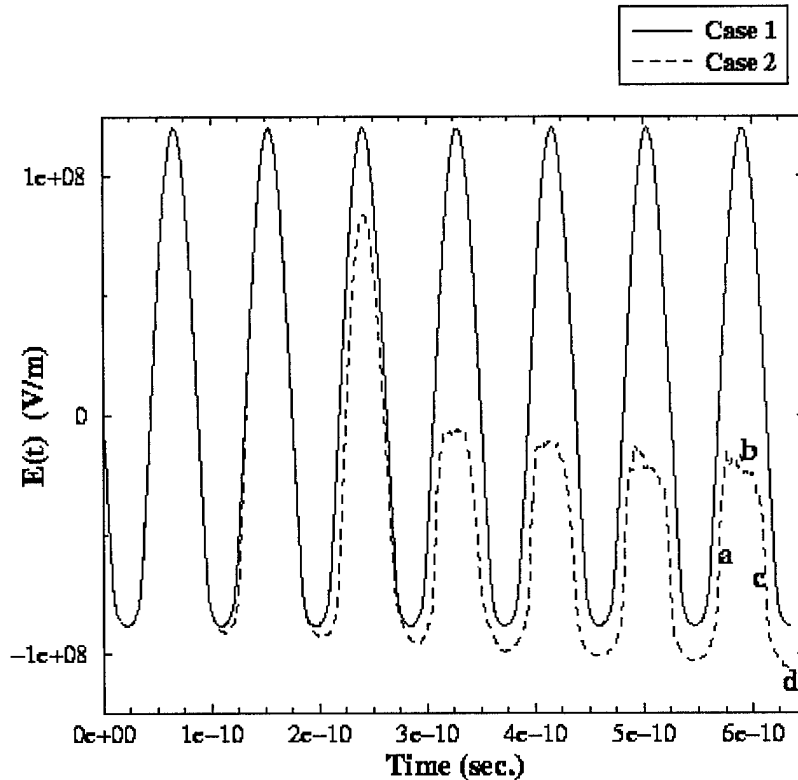


Figure 29: Electric field at emitter vs. time for cases 1 and 2. A time-varying neutral flow is incorporated into this set of simulations by assuming that a monolayer of H atoms is released from the emitter at time  $t = 0$ . Points **a**, **b**, **c**, and **d** on the plot correspond to  $t = 6.5$ ,  $6.75$ ,  $7$ , and  $7.25$  rf cycles respectively.

Points **a**, **b**, **c**, and **d** in Fig. 29 correspond to  $t = 6.5$  rf cycles,  $6.75$  rf cycles,  $7$  rf cycles, and  $7.25$  rf cycles respectively. The following diagnostics (Figures 30-33) show the densities, electric fields, and potentials for cases 1 and 2 at these times.

Let us compare the time evolution of the gas density  $n_g(x)$ , electron density  $n_e(x)$  and ion density  $n_i(x)$  for cases 1 and 2 (Fig. 30 and 31). For case 1, there is no positive ion formation since collisions are turned off. Also, field emission is turned on only for the last  $\frac{1}{4}$  cycle (point **d**) when the field is at its most negative values. For case 2, an electron-ion pair is created for every electron-impact ionization. This increases the electron density and generates ions. Field emission is active over a greater fraction of the rf cycle than in case 1. Also, the field enhancement due to the ions also increases the field emission and leads to greater electron densities.

The enhancement of the field due to the positive ion formation can also be observed in Figures 32 and 33 which compare the fields and potentials of cases 1 and 2 respectively. We know from Eq.'s (1) and (2) that the more negative  $E(x)$  is at the emitter ( $x = 0$ ), the higher the field emission. Figure 32 clearly shows the field enhancement in case 2. In Fig. 33, for case 2, the slope of the potential  $V(x)$  is greater than or equal to zero at the emitter. Thus, for case 2, the slope of an electron's potential energy curve  $U(x) = -eV(x)$  is less than or equal to zero at the emitter, and the electron will tend to fall down a potential hill, enhancing field emission. This increased field emission from the field enhancement leads to the positive feedback loop described above.

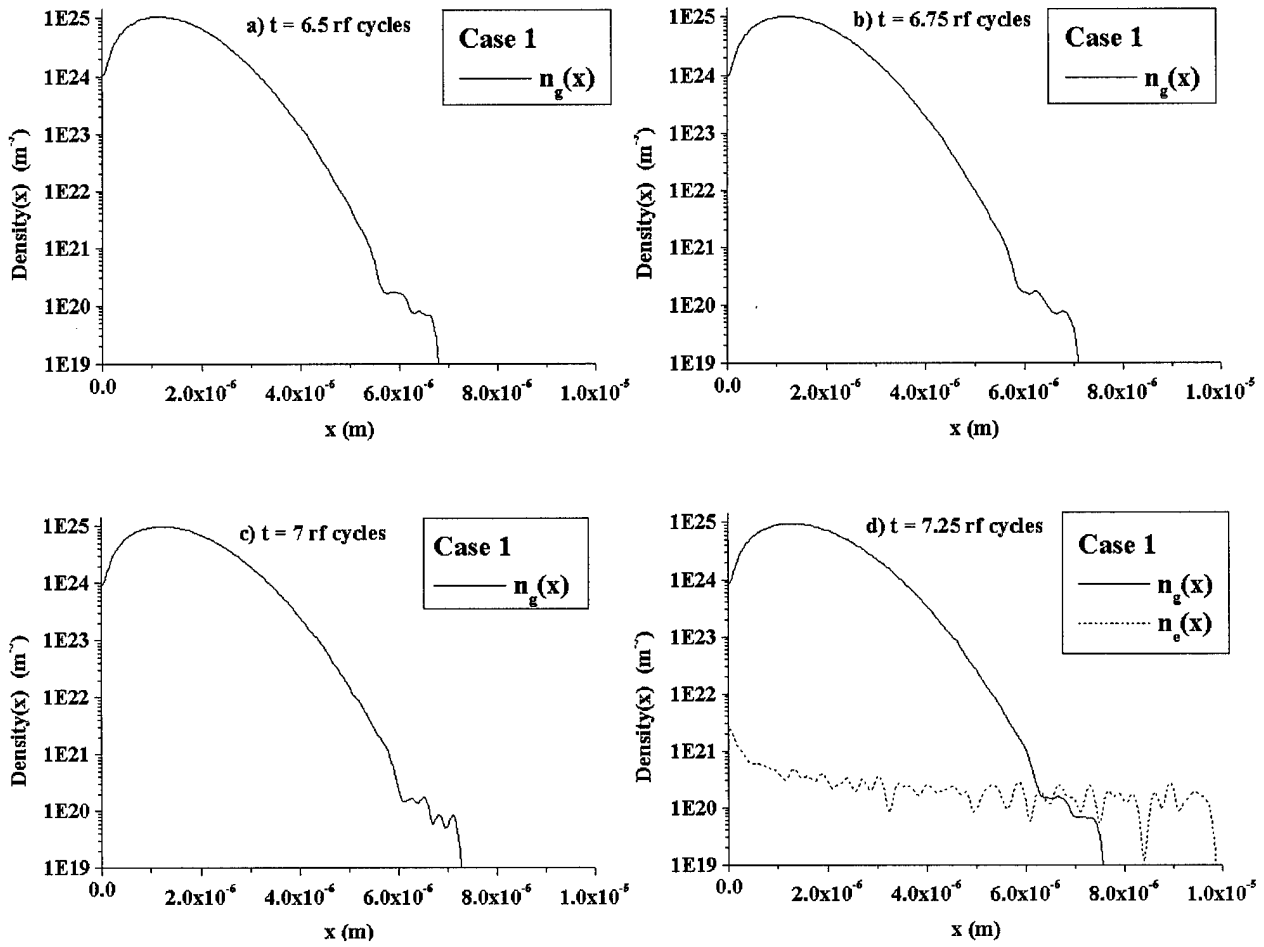


Figure 30: Case 1 densities at  $t = 6.5, 6.75, 7,$  and  $7.25$  rf cycles.

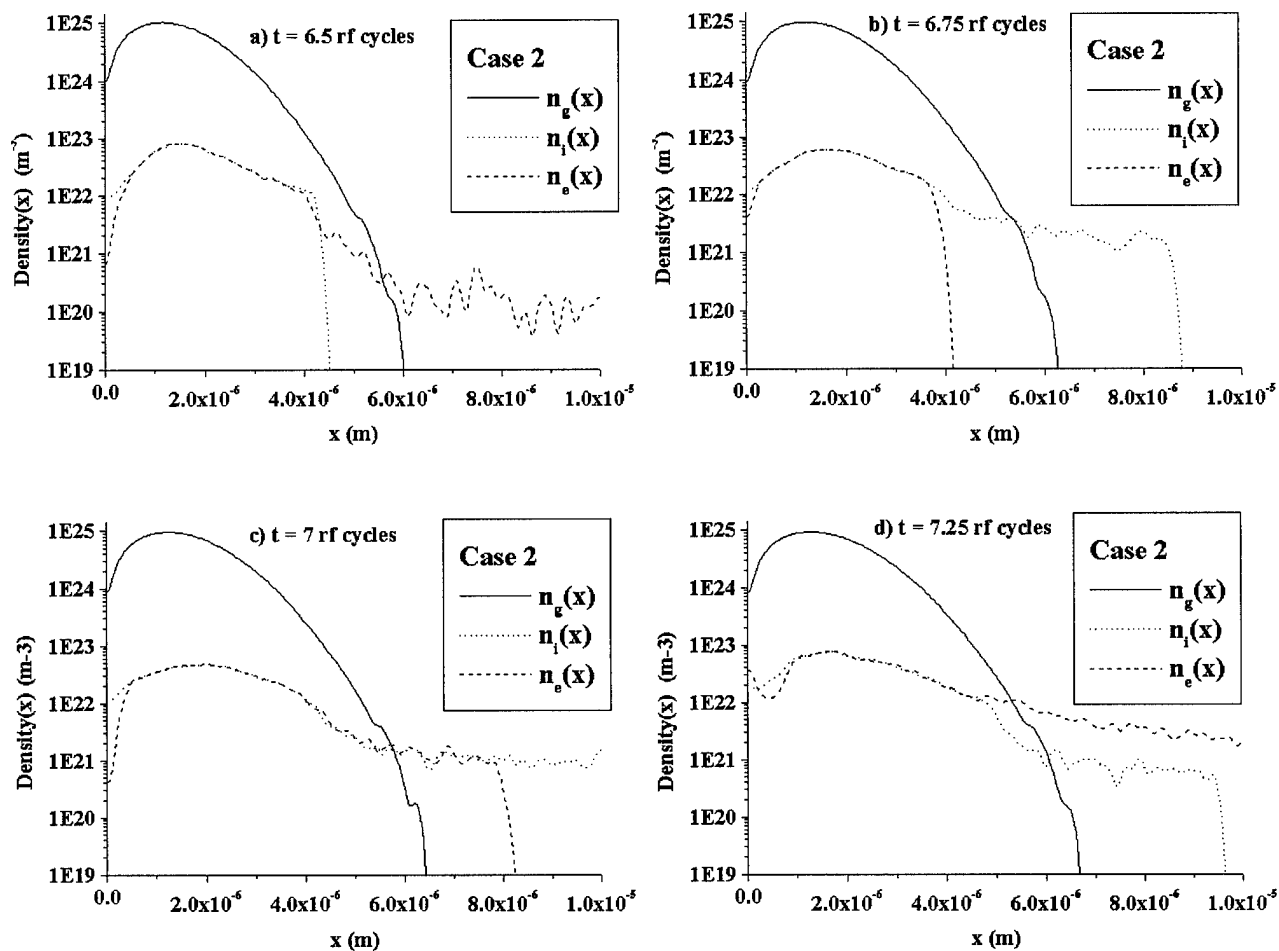


Figure 31: Case 2 densities at  $t = 6.5, 6.75, 7,$  and  $7.25$  rf cycles.

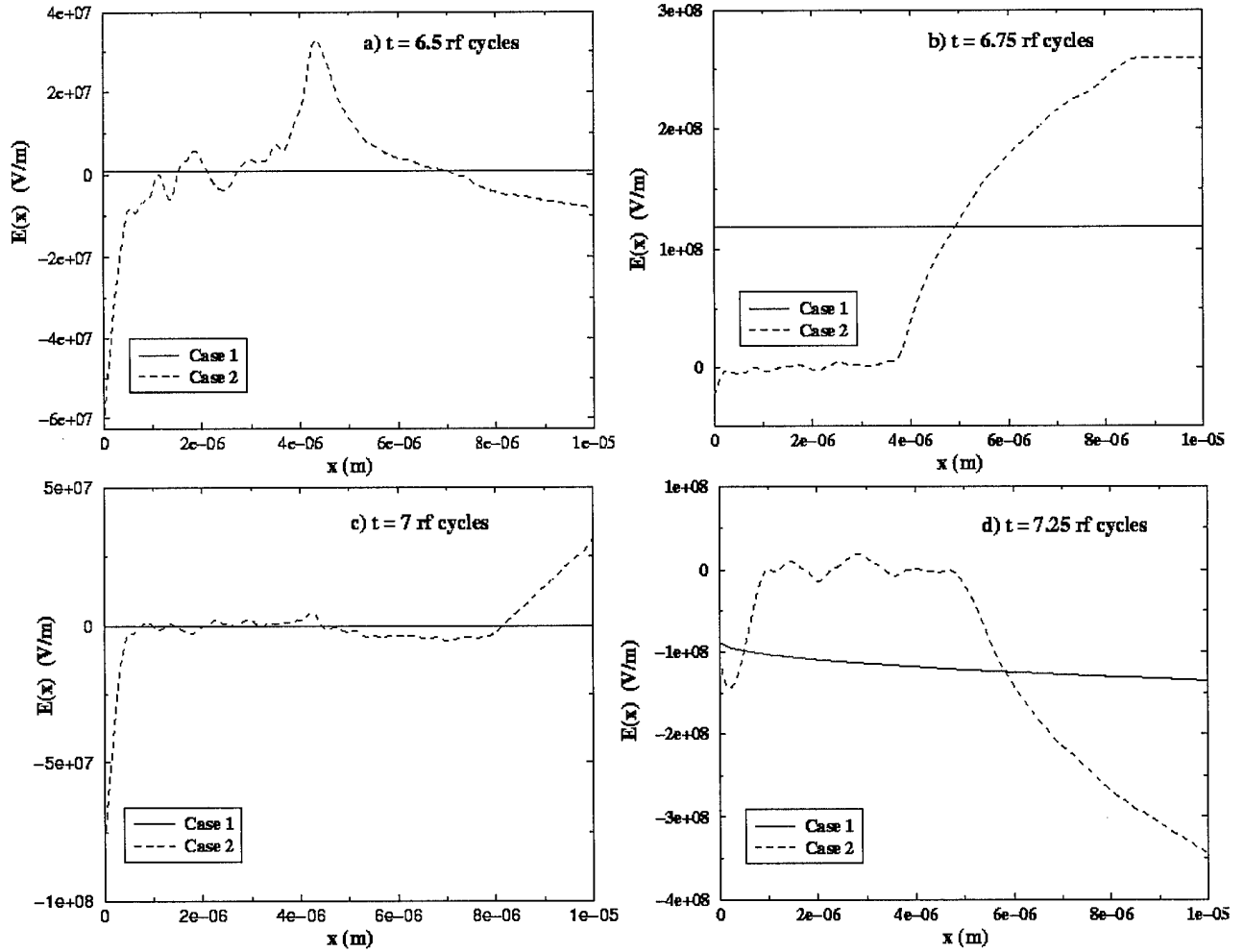


Figure 32: Electric field  $E(x)$  for case 1 (solid line) and case 2 (dashed line) at times  $t = 6.5, 6.75, 7,$  and  $7.25$  rf cycles. Note the field enhancement at the emitter ( $x = 0$ ) in case 2.

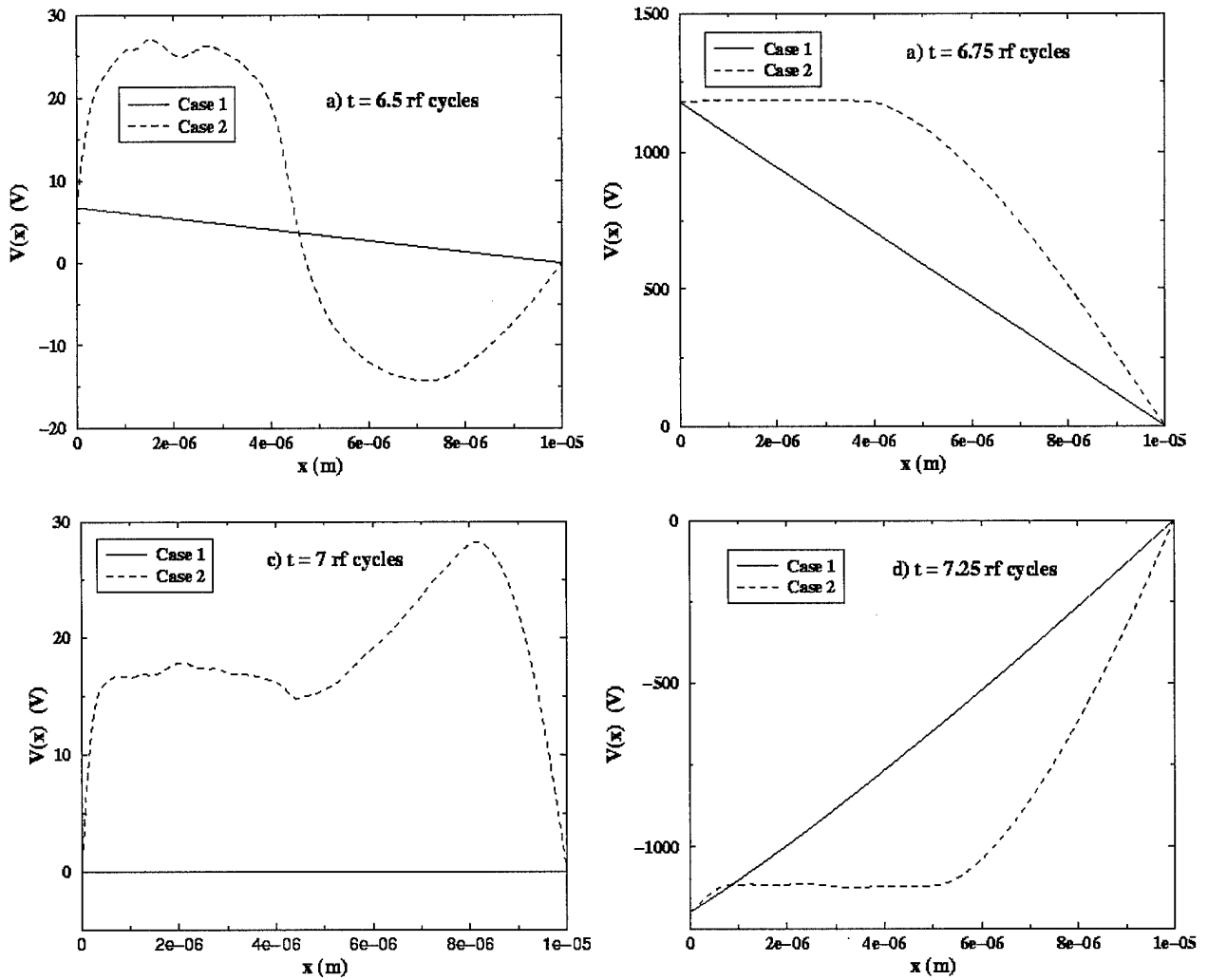


Figure 33: Potential  $V(x)$  for case 1 (solid line) and case 2 (dashed line) at times  $t = 6.5$ , 6.75, 7, and 7.25 rf cycles.

### 2.3.7 Improved neutral flow model

So far our neutral flow model consists of a suddenly released monolayer of gas atoms. However, the actual desorption of neutrals from the surface would be more complicated. We can attempt to model this desorption in two steps. First, we must determine the emitter temperature as a function of time due to heating from the field emission current. Then, we must determine the desorption of atomic H from a copper emitter as a function of the emitter temperature.

To determine the temperature of the emitter, we may model it as a one-dimensional semi-infinite slab. Then, from the heat equation, we have,

$$\frac{\partial^2 T(x,t)}{\partial x^2} - \frac{\rho c_v}{K} \frac{\partial T(x,t)}{\partial t} = -\frac{Q(x,t)}{K} \quad (4)$$

where  $T(x,t)$  is the absolute temperature,  $Q(x,t)$  is the heat source (in our case, the Joule heating from the field emission current),  $K$  is the thermal conductivity,  $c_v$  is the heat capacity, and  $\rho$  is the density of the material. All the coefficients in the heat equation depend on  $T$  so that the equation is non-linear. However, for simplicity, we can use a linearized equation in which the coefficients are evaluated at some  $T=T_0$ .

The boundary and initial conditions for the semi-infinite cathode are as follows: (i) The initial temperature is  $T_0$ , or  $T(x,0) = T_0$ . (ii) No temperature rise is experienced at the far end of the medium so that  $T(\infty, t) = T_0$ . (iii) The temperature gradient at the interface matches that resulting from the heat source, or  $\partial T(0,t)/\partial x = - (1/K) \partial Q(0,t)/\partial x$ .

Let us assume that the H atoms are physisorbed on the Cu surface. To determine the rate of desorption of  $\alpha$  neutrals/m<sup>2</sup> from a homogeneous surface in the event that none is returning from the gas phase, we may assume first order desorption [4]:

$$\frac{d\alpha}{dt} = -\frac{\alpha k_B T}{h} \exp\left(-\frac{E_d}{k_B T}\right) \quad (5)$$

where  $E_d$  is the activation energy for desorption,  $k_B$  is the Boltzmann constant, and  $h$  is Planck's constant.

Once we have incorporated the neutral flow into our model, we can further refine the model by including the effects of ion bombardment on the emitter. We expect the ion bombardment to further heat the surface and lead to further neutral desorption.

### 2.3.8 Empirical data

Many of the parameters needed to model field enhancement depends exactly on how the cavity was prepared. For example:

- Were the metal surfaces of the chamber polished. If so, how?
- Were the chamber walls baked during pump down? If so, what was the temperature duration of the bake out?

The hydrogen concentration in the copper and the hydrogen outgassing rate will depend crucially on this history.

The surface conditions also influence the field emission of electrons. For example:

- Surface defects and grain boundaries may alter the work function of the metal and affect field emission rate. They may also affect the hydrogen outgassing rate.
- Without a better understanding of the geometric field enhancement factor  $\beta$ , a quantitative analysis of field enhancement would be difficult.

Time resolved data would be useful in testing any model. Examples of such data would be

- The temperature of an emitter surface vs. time
- The surface coverage of H atoms on Cu vs. time
- The current vs. time

- The outgassing rate of hydrogen atoms vs. time
- The ion bombardment rate vs. time.
- The electron, ion, neutral densities in the cavity vs. time.

It is advisable to gather more detailed empirical data on the field enhancement problem before embarking on a full-scale model.

### 2.3.9 Conclusion

By using particle simulations, we demonstrated that large positive ion densities can develop near an emitter when field emitted electrons collide with desorbing neutrals. We also showed that the positive ions enhance the field at the emitter. In our first model, we used a constant and uniform neutral background. Then, we incorporated neutral flow into our model by assuming that a monolayer of atoms was released from the emitter surface at the start of the simulation. In order to use more sophisticated neutral desorption models, we need a better understanding of the emitter surface physics. This may be gained by gathering time-resolved empirical data such as the outgassing rate, ion bombardment rate, temperature, densities, and surface coverage at the emitter.

### 2.3.10 References

- [1] C. K. Birdsall, "Particle-in-Cell Charged-Particle Simulations, Plus Monte Carlo Collisions with Neutral Atoms, PIC-MCC", *IEEE Trans. on Plasma Science* **19**(2), pp. 65-85, (1991).
- [2] J. W. Wang and G. A. Loew, "Field Emission and RF Breakdown in High-Gradient Room-Temperature Linac Structures", SLAC center, Stanford University, not yet published.
- [3] J. Knobloch, "Advanced Thermometry Studies of Superconducting RF Cavities", Ph. D. Thesis, Cornell University, (1997).
- [4] P.A. Redhead, J.P. Hobson, and E.V. Kornelsen, "The Physical Basis of Ultrahigh Vacuum", Chapman and Hall Ltd., (1968).

## 2.4 Klystrino Modeling

A high voltage low gradient extended interaction extractor provides a means of extracting microwave power from high voltage devices while avoiding breakdown problems in high voltage gaps. This technique takes the best of klystron performance, with gain from input cavities or other bunching modulation mechanism, and then output from traveling-wave, "extended-interaction", circuits, which extract power over a region rather than from one cavity gap. The extractor may have use in many devices with strongly modulated beams. The extended interaction extractor can prove beneficial to many high voltage, modulated current beam devices, such as relativistic klystrons.

The precedent for this is both theory and early experiments. More recent work has been performed at SLAC, with very deep current modulation (for example, 1.7) fed into a coupled-cavity slow-wave extractor, and high efficiency (over 50%). This information is from R. Phillips,

at SLAC, who also pointed out to us some of the considerations to be taken into account in the extractor, with both tapered velocity and tapered impedance (such as equal voltages per gap).

Modeling this device requires two initial steps. First, a two-dimensional axisymmetric model employing a high voltage strongly modulated beam would provide initial proof of concept. Next, a full three-dimensional model would be required to model realistic power extractors, including multiple waveguides and non-symmetric outputs in multiple cavities. The primary modeling tool is the XOOPIE code, including a 3d extension to the existing 2d code as described above. Final steps may include modeling a complete device, from gun to initial bunching cavities to extractor and collector.

A promising high energy microwave device, proposed by Caryotakis et. Al. at Stanford University, is the klystrino. Also called modular klystrons due to the stackable capability, with quasi-optical couplers, to sum a number of outputs to achieve high energy microwave power. Each individual klystrino module consists of a beam passing through a circuit created using microfabrication techniques. The beam is PPM focused, resulting in light, compact modules. The present design calls for each module to produce 125 kW with a 120 kV beam at 2.5 A.

#### 2.4.1 Klystrino Parameters

Beam voltage and current, beam tunnel diameter and gap transit angles are chosen so that the required beam convergence is not excessive, the focusing magnetic field is relatively low, and most particularly, so that the ratio of plasma wavelength to the PPM period is low and consistent with the cavity spacing. (Cavities are placed between magnet pole-pieces). The resulting parameters are listed below:

Frequency	94 GHz
Beam Voltage	120 kV
Beam Current	2.5 A
Tunnel diameter	0.8 mm
Beam diameter	0.5 mm
Gap transit angles	0.5 mm
Cavity spacing	Approximately 10 mm (for gain cavities)

As a consequence of the above choices, the gun convergence is 81 (15 A/cm<sup>2</sup> cathode current density), the micro-perveance is 0.06, Brillouin field 2700 Gauss,  $\lambda_p/L$  (ratio of plasma wavelength to magnetic period) approximately 5, and cavity beam coupling coefficients about 0.7.

#### 2.4.2 2D Model in XOOPIE

The initial 2D model in XOOPIE reflects the parameters given above, and is performed in cylindrical coordinates, symmetric about the  $r=0$  axis as shown. The figure shows only two of four cavities which are present in the simulation model.

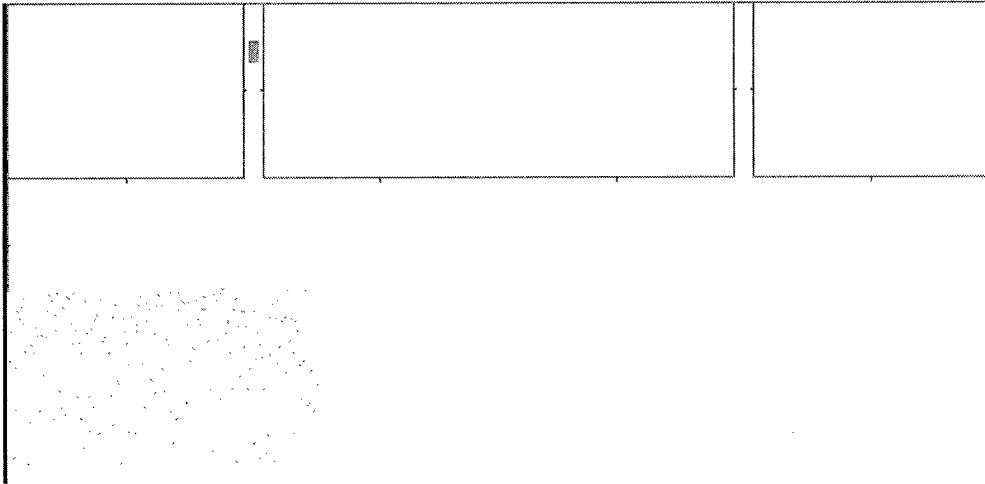


Figure 1 Two-dimensional klystron model, in r-z coordinates.

The confining magnetic field is initially modeled by a homogenous axial magnetic field. This field will later be modified to include the actual fields due to the periodic permanent focussing magnets.

The objectives of the 2D model are to benchmark XOOPIC against previous models simulated in MAGIC by University of California Davis and Stanford University personnel. In addition, insight gained in the rapidly converging two-dimensional model will provide a basis for the three-dimensional model required to study the full physics of the W-band klystron.

The present status of the 2d model is partially complete. While the model is functional, the power predicted by the present design is poor. A redesign of the device is in progress, to be presented at the IEEE ICOPS.

### 2.4.3 3D Model in XOOPIC

The figure, from Caryotakis et al., 1998 MURI-West Annual Report, shows the 3D configuration of the klystrino as it was simulated in Mafia. The parameters are the same as the 2D model: however, the geometry more accurately reflects the shape of the cavities and beam drift tube.

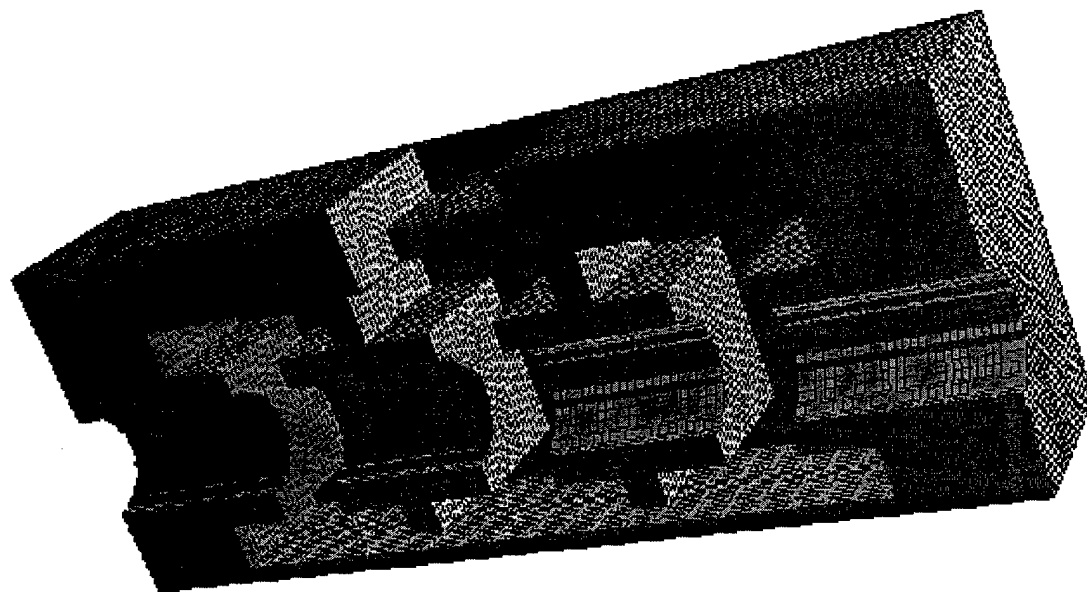


Figure 2 Three-dimensional model of the klystrino (from Caryotakis et al., Stanford University)

The objectives of the 3D simulation are to predict the actual device behavior and ideally to help with design: output power, beam trajectory, power conversion efficiency, and frequency are all of interest.

The 3D model will require approximately 20,000 computational cells, and on the order of 2 million particles for an adequate numerical model. Ten transit times will require ten thousand time steps to acquire the desired data. This amount of computation requires 2 days of CPU time on a fast workstation, so iterative design using a parallel code is conceivable.

3. The 3d Klystrino model is currently incomplete. The model will be presented at the IEEE ICOPS in June, 2000. In particular, the model awaits the redesign of the 2d model. Upon successful simulation of the 2d model, the 3d model can move forward using those results.

---

<sup>1</sup> J. P. Verboncoeur, A. B. Langdon, and N. T. Gladd, "An object-oriented electromagnetic PIC code", *Comp. Physics Comm.* **87**, 199 (1995).

<sup>2</sup> R. A. Kishek and Y. Y. Lau, "Multipactor Discharge on a Dielectric", *Physical Review Letters* **80**, 193-196 (1998).

<sup>3</sup> R. A. Kishek, Y. Y. Lau, L. K. Ang, A. Valfells and R. M. Gilgenbach, "Multipactor discharge on metals and dielectrics: Historical review and recent theories", *Physics of Plasmas* **5**, 2120-2126 (1998).

<sup>4</sup> L. K. Ang, Y. Y. Lau, R. A. Kishek and R. M. Gilgenbach, "Power Deposited on a Dielectric by Multipactor", *IEEE Transactions on Plasma Science*, accepted June (1998).

**REPORT DOCUMENTATION PAGE**

'88

Public reporting burden for this collection of information is estimated to average 1 hour per response including gathering and maintaining the data needed, and completing and reviewing the collection of information, including suggestions for reducing this burden, to Washington Headquarters, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Project, Washington, DC 20503.

AFRL-SR-BL-TR-00-

0217

ig data sources,  
or aspect of this  
1215 Jefferson  
20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE		01 May 96 to 30 Apr 00 Final	
4. TITLE AND SUBTITLE Advanced Microwave Tube Computer Experiments Research				5. FUNDING NUMBERS 61102F 2301/EX	
6. AUTHOR(S) ProfessorBirdsall					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of California 336 Sproul Hall Berkeley, CA 94720				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NE 801 North Randolph Street Rm 732 Arlington, VA 22203-1977				10. SPONSORING/MONITORING AGENCY REPORT NUMBER  F49620-96-1-0154	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION AVAILABILITY STATEMENT APPROVAL FOR PUBLIC RELEASED; DISTRIBUTION UNLIMITED				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  As one of the pioneering efforts in object-oriented (OO) scientific programming, XLOOPIC has demonstrated conclusively the benefits of the methods. The code extensibility and development costs are lower, and the development more rapid than traditional methods. Similar methods of simplifying development and maintenance are now propagating into industrial and laboratory codes, including MRC's MAGIC and AF Phillips Lab's ICE-PIC.					
14. SUBJECT TERMS				15. NUMBER OF PAGES	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT  UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE  UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT  UNCLASSIFIED	20. LIMITATION OF ABSTRACT  UL		