



3 1293 01028 9365

THE 318

LIBRARY
Michigan State
University

This is to certify that the
dissertation entitled

3D3V SIMULATION OF
ELECTRON CYCLOTRON RESONANCE PLASMAS

presented by

Venkatesh P. Gopinath

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical Engineering


Major professor

Date 8/16/94

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
_____	_____	_____
01 28 09	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

MSU is An Affirmative Action/Equal Opportunity Institution

c:\cric\datedue.pm3-p.1

3D3V Simulation of Electron Cyclotron Resonance Plasmas

By

Venkatesh P. Gopinath

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1994

Abstract

3D3V SIMULATION OF ELECTRON CYCLOTRON RESONANCE PLASMAS

By

Venkatesh P. Gopinath

The two and three-dimensional simulation of plasma machines used for materials processing has been investigated. The issues arising when doing self-consistent electromagnetic field/particle type simulations include numerical instabilities, large computer memory requirements and long computer simulation times. Two simulation models were developed to investigate these issues.

For the first model, a self-consistent 3D3V plasma-EM numerical model was developed to simulate discharges inside a compact microwave plasma source. The model coupled an electromagnetic field model and a particle plasma model to simulate plasma production and heating in low pressure ECR plasmas. A grid system was developed to improve numerical stability in cylindrical coordinate system structures. The input parameters to the simulation included operating pressure, gas type and magnetic field pattern. Characteristics of the plasma simulated included plasma temperature, charge density, plasma potential, radial electric field distribution near the walls and power absorption patterns.

For the second model, a 2D3V simulation of the downstream region of plasma processing machines was conducted assuming ϕ symmetry. The plasma source was assumed to provide a constant or spatially variable flux of particles

to the downstream region. The plasma potential and the charge densities across the simulation volume were resolved. A non-uniform particle weighting scheme was implemented to aid stability at the grid points close to the origin.

Simulation models were redesigned to achieve some measure of parallelism. This was done by observing that the movement of particles in particle type plasma simulations are essentially independent of each other once the governing forces have been self-consistently determined. Implementations of 2D3V and 3D3V plasma simulations on a MasPar MP2 massively parallel processor were achieved. Speed-up of over 13 times the fastest locally available serial machine was achieved demonstrating the promise of parallel simulations. Further, the simulation of problem sizes beyond the computing and storage capability of desktop workstations was also demonstrated.

The concept of parallel simulations was modified to spread the computation load across a set of networked desktop workstation machines in a distributed manner. A distributed simulation using PVM was designed to be run across 2, 4 and 8 workstations. The two processor implementation was shown to provide considerable speed up over a single desktop case. However, communication bottlenecks reduced the computational speed-up considerably for the four processor implementation.

**This dissertation is dedicated to my parents
Smt. Ramamani and Sri P.S. Gopinath
for their understanding and support.**

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor Dr. Timothy Grotjohn for his support and guidance throughout this dissertation. I also thank my committee members Drs. Asmussen, Ledford and Reinhard for all their encouragement and for serving on the committee. I would like to thank Dr. Diane Rover for her help in conducting some advanced computations. Last, but in no way the least, I would like to thank my dear wife, Shoba, for all her love and support.

Table of Contents

List of Tables	ix
List of Figures	x
Chapter 1 Introduction	1
1.1 Motivation of 3D3V Simulations	1
1.2 Research Goals	5
1.3 Dissertation Outline	6
Chapter 2 Background	9
2.1 Plasma Description and Fundamental Equations	9
2.2 Numerical Modeling of Plasmas	12
2.2.1 Particle Solution Techniques	12
2.2.1.1 Overall Algorithm of Particle Simulation	13
2.2.1.2 Particle Movement	15
2.2.1.3 Collision Modeling using Monte Carlo Techniques	17
2.2.2 Fluid Models	22
2.2.3 Hybrid Models	24
2.2.4 Poisson's and Maxwell's Equations Solution	25
2.2.5 Geometrical Considerations	31
2.3 Analytical Modeling of Plasmas	35
2.3.1 Schottky Diffusion Model	37
2.3.2 Free-Fall Diffusion Model	38
2.3.3 Ambipolar Diffusion Model	40
2.3.4 Limits of Analytical Models	42
2.4 Vector, Parallel and Distributed Processing Methods	43
2.5 Review of Previous ECR Plasma Modeling Work	47

Chapter 3	Description of Microwave Plasma Sources	50
3.1	Introduction	50
3.1.1	Downstream Plasma Chamber	50
3.1.2	Experimental Methods	53
3.1.2.1	Sample Preparation and Processing	53
3.1.3	Experimental Results of Uniformity and Sputter Rate	53
3.1.4	Summary of Downstream Measurements	65
3.2	Compact Ion Source	66
Chapter 4	A 3D3V Plasma Simulation System	69
4.1	Introduction	69
4.2	Design of the Plasma Simulation	69
4.2.1	A Non-Terminated Grid Design	70
4.2.2	Poisson's Equation Solution	74
4.2.3	Plasma Kinetic Model	76
4.2.3.1	Particle-Neutral Collisions	76
4.2.3.2	Charged Particle-Particle Collisions	78
4.2.3.3	Electron-Neutral Ionizing Collisions	79
4.2.4	Maxwell's Equation Solution	81
4.2.5	Self-Consistent 3D3V Plasma Simulations	82
4.2.5.1	Geometrical Requirements	82
4.2.5.2	Combined Plasma-Maxwell Simulation	84
4.2.5.3	Power Absorption Calculations	87
4.3	Efforts to Speed up the Simulation	88
4.3.1	Cycling the EM Program	88
4.3.2	Implementation of a Parallel EM-Plasma Version	93
Chapter 5	Downstream Plasma Simulations	99
5.1	Introduction	99
5.2	2D3V Downstream Simulation	100
5.2.1	An Unequal Weighting Scheme	102
5.2.2	Poisson's Equation Solution	106
5.2.3	Example Simulation	108

Chapter 6	Simulation of a Compact ECR Source	120
6.1	Simulation of a Compact Ion Source	120
6.2	Results	123
Chapter 7	Implementation on the MasPar MP2 System	131
7.1	Introduction	131
7.2	Implementation of a Two Dimensional Parallel Program	134
7.2.1	Movement of Particles Across PEs	134
7.2.2	Poisson's Equation Solution	135
7.2.3	Calculation of Fields	135
7.2.4	Results of the Two Dimensional Simulation	135
7.3	Implementation of a 3 Dimensional Parallel Program	139
7.3.1	Implementation of Third Dimension in Memory	139
7.3.2	Implementation of Third Dimension Across PEs	142
7.3.2.1	Movement of Particles Across PEs	143
7.3.2.2	Poisson's Equation Solution	145
7.3.2.3	Calculation of Fields	145
7.4	Implementation of 3D Maxwell's Equations Solution	147
7.5	Discussion of Results	148
Chapter 8	Distributed 3D3V Simulation	153
8.1	Introduction	153
8.2	Parallel Virtual Machine	153
8.3	A Distributed Poisson's Equation Solver	154
8.4	Distributed 3D3V Plasma Simulation Implementation	160
8.5	Discussion of Results	163
Chapter 9	Conclusion	170
9.1	Summary of Research	170
9.2	Directions for Future Research	173
References		175

List of Tables

Table 3.1: Sputtering rate and uniformity vs. microwave power	56
Table 3.2: Sputtering rate and uniformity vs. pressure	56
Table 3.3: Sputtering rate and uniformity vs. RF induced bias	57
Table 3.4: Sputtering rate and uniformity vs. downstream distance	57
Table 7.1: Specification of the three systems	140
Table 7.2: Results for the 2D simulation	140
Table 7.3: Results for the 3D simulation	149
Table 8.1: Specification of the two systems	165
Table 8.2: Results of distributed simulation	165
Table 8.3: Results on a sub-netted system	167

List of Figures

Figure 1.1: An example plasma processing system.	2
Figure 2.1: Time evolution of particles during a simulation.	14
Figure 2.2: Selection of collision event by Monte Carlo method.	19
Figure 2.3: Flow chart of a typical electrostatic plasma simulation.	20
Figure 2.4: Distribution of particle charge to grid.	28
Figure 2.5: Cylindrical grids with uniform grid spacing.	32
Figure 2.6: Cylindrical grids with non-uniform grid spacing.	34
Figure 2.7: A 25 cm diameter ECR system.	41
Figure 2.8: Example of symmetry in a plasma simulation.	44
Figure 3.1: Comparison of theory with experiment for 9 cm system.	55
Figure 3.2: Oxide etched vs. power.	58
Figure 3.3: Oxide etched vs. pressure.	59
Figure 3.4: Oxide etched vs. RF induced bias.	60
Figure 3.5: Comparison of theory with experiment for downstream points.	61
Figure 3.6: Comparison of theory with experiment for 3 downstream points.	63
Figure 3.7: Comparison of theory with experiment for a 6 inch wafer.	64
Figure 3.8: Compact ECR ion source.	67
Figure 4.1: Non-terminated grid scheme.	72
Figure 4.2: Elementary cell in non-terminated scheme.	73
Figure 4.3: Example ionization cross section plot.	80
Figure 4.4: Combined plasma-EM simulation flowchart.	86
Figure 4.5: Time evolution of fields for the serial and cycled cases.	90
Figure 4.6: Energy of particles with source cycled on and off.	91
Figure 4.7: Number density of particles with source cycled on and off.	92
Figure 4.8: A parallel EM-plasma implementation.	94
Figure 4.9: Time evolution of fields for the serial and parallel cases.	95

Figure 4.10: Evolution of number density for the serial and parallel cases.	96
Figure 4.11: Time evolution of energy for the serial and parallel cases. . .	97
Figure 5.1: Downstream simulation zone.	101
Figure 5.2: Unequal weighting implementation.	104
Figure 5.3: Implementation of Von Neumann condition.	107
Figure 5.4: Time evolution of number of particles.	109
Figure 5.5: Plasma potential in the simulation region.	111
Figure 5.6: Time evolution of particle energy.	112
Figure 5.7: Electron density vs. z	113
Figure 5.8: Electron density vs. r	114
Figure 5.9: Ion density vs. r	115
Figure 5.10: Ion density vs. z	116
Figure 5.11: A Bessel function type flux distribution.	117
Figure 5.12: Electron density vs. z with non-uniform source.	118
Figure 6.1: B field contours inside ion source.	121
Figure 6.2: Structure being simulated.	122
Figure 6.3: Time evolution of average particle energy.	124
Figure 6.4: Power absorption locations	125
Figure 6.5: Time evolution of number of particles.	126
Figure 6.6: Plasma potential in the rz plane.	128
Figure 6.7: Time average values of E_r along the edge.	129
Figure 6.8: Electron energy distribution function.	130
Figure 7.1: Solving Poisson's equation.	136
Figure 7.2: Solving for electric field.	137
Figure 7.3: Interpolation of x component of E-Field at particle location. . .	138
Figure 7.4: Implementation of 3 rd dimension in memory.	141
Figure 7.5: Implementation of 3 rd dimension across PEs.	144
Figure 7.6: Motion of particles across PEs in the z and xy planes.	146
Figure 8.1: Distribution of load across computers.	156
Figure 8.2: Shared planes across computers.	157
Figure 8.3: A discarded partitioning scheme.	158
Figure 8.4: Flow chart of distributed Poisson's equation solver.	161
Figure 8.5: Flow chart of distributed 3D3V plasma simulation.	162
Figure 8.6: Comprehensive view of results.	169

Chapter 1

Introduction

1.1 Motivation of 3D3V Simulations

Electron Cyclotron Resonance (ECR) plasmas have demonstrated excellent **potential** for semiconductor processing because of their ability to create high **densities** of charged species and free radicals without high plasma sheath **potentials** and contamination from electrodes [1]. Unlike RF parallel plate **reactors**, ECR plasma generation is isolated from wafer biasing so that ion flux **and** ion energy to wafer surfaces can be independently controlled. Since ECR **plasma** reactors operate at lower pressures [2], the mean free path is very long **leading** to anisotropic ion transport across the sheath. High power absorption **densities** and magnetic field confinement help to obtain large etching and **deposition** rates at lower pressures by maintaining large charged particle **densities**.

A typical plasma source detailing the plasma generation and downstream **regions** is shown in Figure 1.1. The plasma is generated in the region under the **quartz** dome, termed the source region, with the help of the ECR magnets. It **then** flows into the processing area, also termed the downstream region. **Permanent** magnets in the downstream region aid in the confinement of charged **particles**.

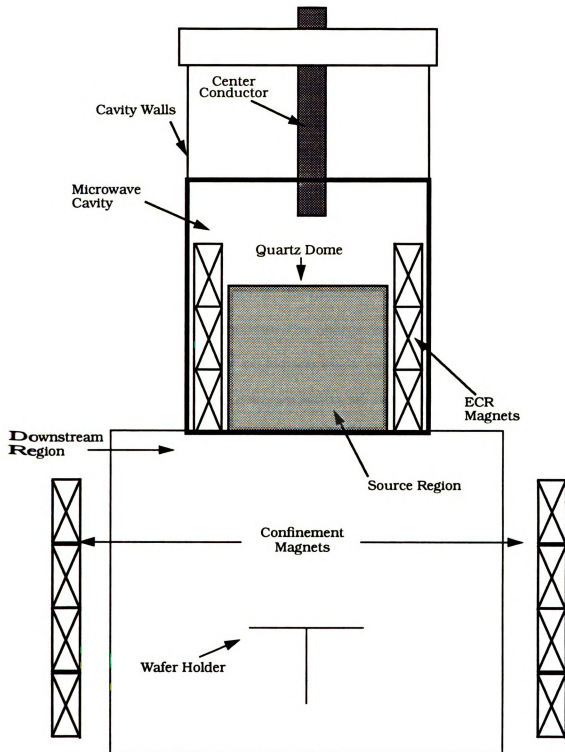


Figure 1.1: An example plasma processing system.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

To gain a better understanding of microwave plasma source operation, the kinetics of plasma excitation and charged particle transport need to be studied. In earlier studies it has been demonstrated using simple models ([3]-[5]), that the plasma sputter rate due to argon ions bombarding a surface is strongly dependent on the incident ion density and energy of the ions when they arrive at the surface. The ambipolar model for charged particle behavior used in these studies assumed a uniform ion charge density leaving the plasma source and entering the downstream processing region, and an absence of downstream confinement magnets. Simulations to predict plasma properties in the case of nonuniform source charge density, in the presence of confinement magnets and for more complex gases are of interest. The charge density inside a discharge is a complex function of a number of factors such as reactor geometry, form of excitation, pressure, type of gas, presence of magnetic confinement, etc. Therefore, in order to accurately predict the downstream behavior of a plasma discharge, accurate two or three dimensional (3D) simulations which take the above factors into consideration will have to be done.

One of the many ways of creating and maintaining a plasma is by the ECR effect. ECR plasmas are operated at low pressures (0.1mTorr-10mTorr) so that the electron mean free path is large enough for efficient energy coupling. The energy from a time varying electric field can be coupled to an electron very efficiently in the presence of an appropriate static magnetic field [6]. For example, a 2.45 GHz microwave electric field transfers energy to an electron very efficiently in the presence of a static magnetic field of strength 875 Gauss. In the case of low pressure plasmas where collisions are few and volume recombination is negligible, ECR can be used to create and maintain a high density discharge.

The static magnetic field has the additional advantage of confining the charged particles inside the discharge leading to higher charge densities or alternatively cooler electron temperatures. Three dimensional simulations of microwave cavities used for plasma generation present the possibility of predicting the effect of cavity shape, permanent magnet strength, size and orientation, choice of gas and other operating parameters on the final plasma properties without having to build prototypes.

While 1D and 2D ([8]-[16]) simulations of discharges in the source and downstream locations have been done extensively, it is only recently that these techniques are being extended to 3D structures ([17]-[21]). Three-dimensional-three-velocity (3D3V) particle simulations of plasmas account for the behavior of charged species in the three geometrical dimensions and the three velocity directions of particle motion. The fundamental restriction on grid based particle simulation of discharges is that the grid size should be approximately the Debye length. Consider a 3D3V simulation of a small cylindrical discharge of radius 1.5 cm and length 1.5 cm. Assuming a Debye length of 0.05 cm, a grid of approximately $30 \times 210 \times 30$ (r, ϕ , and z directions) nodes would be required since the Debye length requirement has to be satisfied for the grids on the circumference. A simulation consisting of two types of charged particles and at least eight particles per grid would store six floating point variables per charged particle (location and velocities in each of the 3 dimensions) and at least seven floating point variables per grid point (electric and magnetic fields in the three dimensions and charge density). A simple calculation shows that it would require computer memory in excess of 70 MBytes. Therefore 3D3V simulations will have to be performed either by using computers of sufficient memory and

speed or by modifying the problem to fit the existing computer resources. The latter approach can take many different forms, one of which is to take advantage of any symmetry present in discharges. This is made easy by the fact that most discharges show some form of ϕ independence or repeatable pattern. A second approach is to identify parallel computing components of the simulation and distribute the computation and hence the memory requirements amongst a number of slower machines ([20]-[22]). This approach has the advantage of using available computing resources and distributing the load across several low-cost machines.

1.2 Research Goals

The objective of this study is to implement, test and verify state-of-the-art plasma simulation techniques, software and methodologies for the modeling of discharges in both source and downstream regions of low-pressure, high-density ECR plasma reactors. The primary goal of this study is to predict the three dimensional charged particle fluxes, energies and distributions inside the plasma reactor as a function of the input parameter space which includes microwave power, static magnetic fields, gas composition, flow rate, pressure and reactor geometry. An additional goal is to verify the simulations using measured plasma parameters. Simulation software developed will use and extend state-of-the-art plasma simulation techniques including complex geometries in three dimensions, self-consistent plasma-Maxwell equations solutions, and advanced computing techniques.

This study will develop the capability to simulate plasma discharges in microwave ion/free radical source developed at Michigan State University [24]

using collision cross sections and ionization rates from the literature. Also, this study will develop the simulation tools to model plasma behavior in downstream processing regions of microwave plasma sources. Additionally, this research will also develop software to simulate reactor structures in a number of different coordinate systems including cartesian and cylindrical.

Since self-consistent plasma-Maxwell simulations are very compute and memory intensive, it is necessary to seek computational methods and options which lead to solutions in a reasonable amount of time. In the case of simulating large geometries, parallel and distributed computing approaches are to be considered.

1.3 Dissertation Outline

This dissertation is divided into four subjects: literature review of plasma simulations; development of numerical plasma simulation model, comparison of analytical models, numerical models and experimental data, and the extension of these techniques to use massively parallel processor (MPP) computers and distributed computing.

Chapter 2 is a multi-part review of plasma fundamentals and different models used in describing plasma behavior. The first part introduces the fundamental equations that describe and control the behavior of plasma. The second segment introduces the fundamentals of numerical modeling of plasmas including different methods like, i.e., fluid and hybrid methods of plasma simulation. A review of the geometrical considerations involved and the role they play in plasma simulations is also introduced. The third part describes various analytical models used in describing plasma behavior like the free-fall diffusion

and ambipolar diffusion models and their limitations. The fourth section of this chapter introduces the many different ways in which time consuming simulations can be speeded up with the help of advanced computing techniques. Lastly, a review of previous ECR plasma modeling is introduced highlighting the need for the research done in this dissertation.

Chapter 3 describes the two experimental systems studied and modeled. Additionally, the application and limitation of a simple ambipolar diffusion model to predict/model the sputtering rate and uniformity of oxide removal from silicon wafers is described. The aim of this exercise is (1) to correlate between the analytical ambipolar model of charge density and the resulting sputter characteristics under different operating conditions and two different wafer sizes, and (2) to establish the limitations of simple analytical models and the need for numerical models.

Chapter 4 introduces the 3D3V plasma simulation techniques developed in this research. The particle type simulation details developed for this research are explained. This chapter also illustrates the non-terminated cylindrical grid methods developed in order to improve numerical stability. The different collisions modelled and the methods used to model these are also presented. The use of different techniques to speed up the simulations are also discussed.

A two dimensional implementation of the particle model to the simulation of downstream plasma dynamics is presented in Chapter 5. A comparison of the results of this model with the ambipolar theory is done.

Chapter 6 applies the complete 3D3V plasma-Maxwell's equations solution to the simulations of an ion source used at Michigan State University. Results of the simulations are compared with experimental characterizations of the source.

The use of massively parallel processors as a useful tool to speed up plasma simulations is explained in Chapter 7. This chapter details the various techniques used to speed up plasma and EM simulations in two and three dimensions. This chapter also shows the possibility of simulating structures and problem sizes beyond the capability of conventional desktop computers.

Chapter 8 demonstrates the possibility of using many conventional desktop machines to achieve considerable speed-up over single machine simulations. The concept of distributed applications and the methods used to achieve these will also be presented.

Finally a summary of important results of this research and directions for future research is presented in Chapter 9.

This work was performed under the guidance of Dr. T. A. Grotjohn, Associate Professor in the Department of Electrical Engineering, Michigan State University. The research in this dissertation has been published in part in refereed journals and presented at international conferences ([5],[25],[26],[27] and [74]).

Chapter 2

Background

2.1 Plasma Description and Fundamental Equations

Plasmas are composed of particles including ions, electrons and neutrals which are excited by the input power source through a variety of heating and collision processes. In a plasma, the electrons and ions no longer behave as separate entities but exhibit a collective behavior. Plasma behavior is governed by plasma kinetics and the electric and magnetic fields. Therefore, plasma kinetics must be solved in conjunction with either Maxwell's equations or, under electrostatic conditions with Poisson's equation only. The plasma then interacts with a material immersed in it leading to sputtering, etching or deposition [28] as the case maybe.

The Maxwell's equations are [29]

$$\nabla \times \mathbf{E} = -\mu \frac{\partial}{\partial t} \mathbf{H} \quad (2.1)$$

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial}{\partial t} \mathbf{E} + \mathbf{J} \quad (2.2)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.3)$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon} \quad (2.4)$$

where \mathbf{E} is the electric field strength, \mathbf{H} is the magnetic field strength, \mathbf{J} is the electric current density, \mathbf{B} is the magnetic flux density, ϵ is the electric

permittivity of the medium, μ is the magnetic permeability of the medium and ρ is the electric charge density. Equation (2.3) can be solved separately to calculate the static \mathbf{B} field from the permanent magnets encircling the cavity. In the case of the source region, these magnets will provide the ECR zones and in the downstream case, if present, will provide charge particle confinement. In the presence of just a static \mathbf{E} field, equation (2.4) can be modified to obtain Poisson's equation as

$$\nabla^2 V = -\frac{\rho}{\epsilon} \quad (2.5)$$

where

$$\mathbf{E} = -\nabla V \quad (2.6)$$

and V is the electrostatic potential.

In addition to the above equations, plasma kinetics are described in the form of a general equation that describes the temporal and spatial variation of relevant macroscopic variables [6]. This general equation (2.7) known as the Boltzmann equation can be written for a species as follows

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \mathbf{a} \cdot \nabla_{\mathbf{v}} f_\alpha = \left(\frac{\partial f_\alpha}{\partial t} \right)_{coll} \quad (2.7)$$

where f_α is the distribution function in six dimensional phase space (locations and velocities) for the particle type α , \mathbf{v} is the velocity of the particle, \mathbf{a} is the acceleration, ∇f_α and $\nabla_{\mathbf{v}} f_\alpha$ are the spatial and velocity gradients of the distribution function, respectively. If the particle type under consideration is charged, the acceleration is related to the electric and magnetic fields in equations (2.1) - (2.4) by the Lorentz force equation as follows

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{1}{m} (q\mathbf{E} + q(\mathbf{v} \times \mathbf{B})) \quad (2.8)$$

where q is the charge on the particle and m is its mass. The general Boltzmann equation can be modified to be a set of transport equations of average values for physical variables $x(\mathbf{v})$ (e.g., velocity, energy). This is done by multiplying equation (2.8) by $x(\mathbf{v})$ and integrating over all velocity space. A general transport equation in terms of average values can be derived to be

$$\frac{\partial}{\partial t}(n_\alpha \langle x \rangle_\alpha) + \nabla \bullet (n_\alpha \langle x \mathbf{v} \rangle_\alpha) - n_\alpha \langle \mathbf{a} \bullet \nabla_v x \rangle_\alpha = \int_v x \left(\frac{\partial f_\alpha}{\partial t} \right)_{coll} d^3 v \quad (2.9)$$

where n_α is the total number of particles of type α [6]. Substituting various variables for the general variable $x(\mathbf{v})$ gives a number of useful relations, termed the *moments of the Boltzmann equation*, like the continuity equation, momentum transport equation, energy transport equation, etc. However, the set of transport equations thus derived contain more variables than independent equations. For example, the momentum transport equation contains the pressure dyad as an unknown. Therefore certain simplifying assumptions have to be made in order to truncate the hierarchy of the moments of the transport equation. The simplest of these is the *cold plasma* model wherein only the first two moments of the Boltzmann equation, i.e. the density and average velocity, are considered by assuming that the pressure temperature dyad is equal to zero. If on the other hand, the divergence of the heat flux vector is assumed to be zero, in effect assuming zero thermal flux, one can truncate the series after the third moment, i.e., the energy transport equation, and obtain the *warm plasma* model. Further discussion of simulating plasmas using equations derived in this fashion is made in Section 2.2.2.

2.2 Numerical Modeling of Plasmas

2.2.1 Particle Solution Techniques

Particle type simulation of plasmas attempt to simulate the behavior of a large number ($10^9/\text{cm}^3$ - $10^{11}/\text{cm}^3$) of electrons by considering a few representative particles since it would be quite impossible to perform simulations in a reasonable amount of time if all particles were considered. These representative particles, called *superparticles*, carry the same mass as that of the charged particle they are representing but contribute to the charge density a factor greater than one. For example, if one were to simulate electron motion in a plasma of electron charge density $10^9/\text{cm}^3$ in a chamber of volume 1cm^3 by 10^6 superparticles, each superparticle would carry a weight of 1000 electron elementary charges.

The particles, electrons and one or more type of ions (both negative and positive), are then moved under the influence of static and time varying \mathbf{E} and \mathbf{B} fields. At the end of each timestep iteration, the contribution of the particles to the charge and current densities on the background grids are computed and new fields are calculated by solving both Poisson's and Maxwell's equations or only Poisson's equation as the case demands. Any particle going out of the simulation region is considered lost. The loss of particles to the walls is normally compensated by the generation of new particles. This generation term can take one of two forms. In the case of the plasma excitation region, some electrons in the ECR regions acquire energy higher than the ionization potential of the background gas creating new electron-ion pairs. In the other case of downstream plasma simulations, a charged particle flux equal to the ion flux

emanating from the source region is added as particles during each iteration. A low pressure plasma simulation can be considered to have reached a steady state solution when the rate of loss to the walls is compensated for by the rate of generation as shown in Figure 2.1. Simulations of high pressure plasma should also include volume recombination and collisional (Joule) heating of particles.

2.2.1.1 Overall Algorithm of Particle Simulation

The structure to be simulated is broken up into grids in three dimensions such that the grid size is of the order of the Debye length. The number of particles (electrons and one or more type of ions) is chosen such that there are at least eight particles per grid in order to obtain a numerically stable solution [7]. The input parameters for a typical simulation include neutral gas pressure and type, initial charge density, electron and ion temperatures and cavity dimensions. The temperatures are used to initialize the particle velocities using a Maxwell-Boltzmann distribution. The type of the neutral gas decides the ionization potential and its density determines the particle-neutral collision rates along with the particle densities. The initial charge densities along with the cavity dimensions determine the charge weighting of the superparticles. The location of the particles are randomly chosen in the cavity giving a final uniform initial distribution. The boundary conditions may be of the form of grounded walls and some constant \mathbf{E} field.

Initial charge and current densities are calculated from the above initial location and velocities respectively and are then used to calculate the new \mathbf{E} and \mathbf{B} fields by solving Maxwell's equations. The electrons and ions are moved under

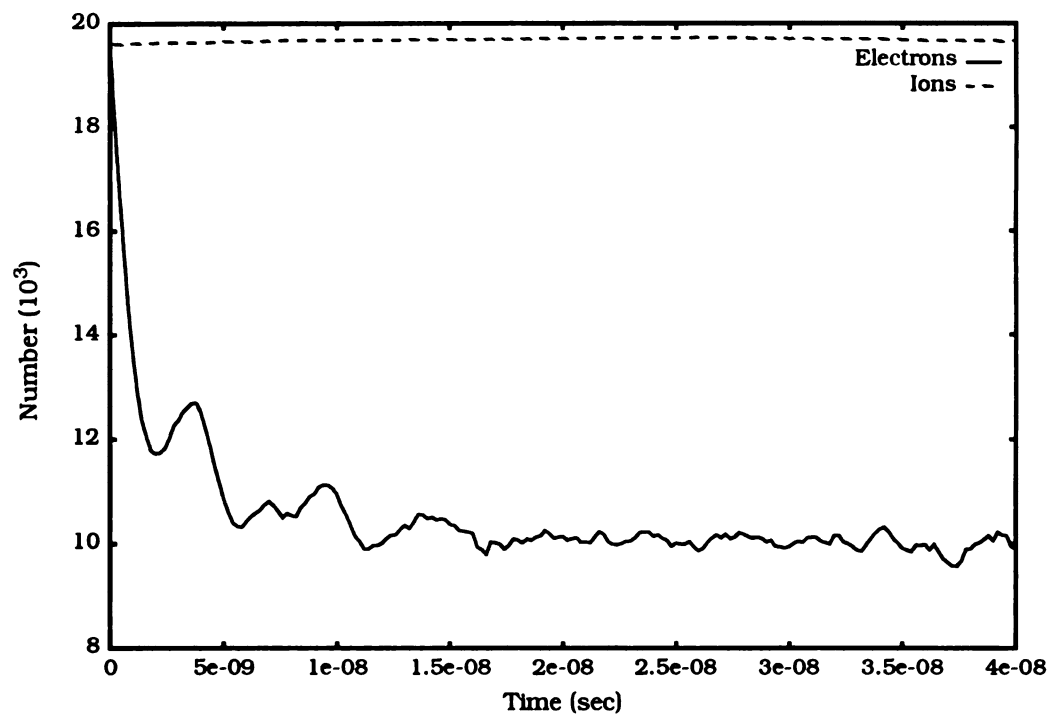


Figure 2.1: Time evolution of particles during a simulation.

the influence of these \mathbf{E} and \mathbf{B} fields and their trajectories are checked to see if they are lost to the walls. Monte Carlo techniques, described in Section 2.2.1.3 below, are used to determine the occurrence of a collision for a particle and appropriate actions are taken. At the end of each iteration, new charge and current densities are calculated and the whole process is repeated again.

In the case of the source region ECR plasma simulations, some electrons acquire energy above the ionizing potential of the background gas and the occurrence of ionization is determined by the ionization cross-section for that gas. In downstream simulations, the input flux compensates for the loss of particles. The simulation is considered complete when the rate of loss is compensated by the rate of gain of particles.

2.2.1.2 Particle Movement

The kinetics of charged particle motion is done by numerically solving the Lorentz force equation, given by equation (2.8), for the timestep Δt along with

$$\frac{d\mathbf{r}}{dt} = \mathbf{v} \quad (2.10)$$

where \mathbf{v} is the velocity and \mathbf{r} is the position of the particle.

In order to solve equations (2.8) and (2.10) numerically and obtain a stable solution, consider a single differential equation of the form

$$\frac{dx}{dt} = f(x, t) \quad (2.11)$$

with the initial condition

$$x(t_0) = x_0 \quad (2.12)$$

where $f(x, t)$ is some single valued function of the variables x and t .

Equation (2.11) can be expanded [30] by Taylor series about the point $t = t_i$ for the time t_{i+1} . By neglecting higher order terms gives, Taylor series the forward Euler algorithm as

$$x_{i+1} = x_i + hf(x_i, t_i) \quad (2.13)$$

where x_{i+1} is the position at $t=t_{i+1}$ and h is the timestep ($t_{i+1} - t_i$).

This simple numerical technique turns out to be non-energy conserving. The solution of these equations in a way that is energy conserving uses the *leapfrog* method. In this method the velocity and position are calculated on an alternating half step division basis.

The charged particle motion using equations (2.8) and (2.10) can be written as

$$\mathbf{v}_{i+1} = \frac{q}{m} \mathbf{E}_{i+\frac{1}{2}} h + \left(\mathbf{v}_{i+\frac{1}{2}} \times \frac{q}{m} \mathbf{B} \right) h + \mathbf{v}_i \quad (2.14)$$

and

$$\mathbf{r}_{i+\frac{3}{2}} = \mathbf{r}_{i+\frac{1}{2}} + h\mathbf{v}_{i+1}. \quad (2.15)$$

The velocity at time $t_{i+1/2}$ is replaced by the average of the velocity at t_i and t_{i+1} and after defining ω_c as $q\mathbf{B}/m$, equation (2.14) can be written in a matrix form as 3 coupled equations

$$\begin{bmatrix} v_{x,i+1} \\ v_{y,i+1} \\ v_{z,i+1} \end{bmatrix} = \frac{q}{m} h \begin{bmatrix} E_{x,i+\frac{1}{2}} \\ E_{y,i+\frac{1}{2}} \\ E_{z,i+\frac{1}{2}} \end{bmatrix} + \frac{h}{2} \begin{bmatrix} v_y \omega_{cz} - v_z \omega_{cy} \\ v_z \omega_{cx} - v_x \omega_{cz} \\ v_x \omega_{cy} - v_y \omega_{cx} \end{bmatrix}_{i+1} + \frac{h}{2} \begin{bmatrix} v_y \omega_{cz} - v_z \omega_{cy} \\ v_z \omega_{cx} - v_x \omega_{cz} \\ v_x \omega_{cy} - v_y \omega_{cx} \end{bmatrix}_i + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}_i. \quad (2.16)$$

After some manipulations, the above equation can be written as

$$\mathbf{v}_{i+1} = \left[\mathbf{I} - \frac{h}{2} \mathbf{w}_c \right]^{-1} \frac{qh}{m} \mathbf{E}_{i+\frac{1}{2}} + \left[\mathbf{I} - \frac{h}{2} \mathbf{w}_c \right]^{-1} \left[\mathbf{I} + \frac{h}{2} \mathbf{w}_c \right] \mathbf{v}_i \quad (2.17)$$

where

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

and

$$\mathbf{w}_c = \begin{bmatrix} 0 & \omega_{cz} & -\omega_{cy} \\ -\omega_{cz} & 0 & \omega_{cx} \\ \omega_{cy} & -\omega_{cx} & 0 \end{bmatrix}. \quad (2.19)$$

2.2.1.3 Collision Modeling using Monte Carlo Techniques

The mathematical techniques used by the Monte Carlo method are based on the selection of random numbers and hence are very well suited for problems which show statistical distributions. In principle, Monte Carlo methods can be considered as a very general mathematical tool for the solution of a great variety of problems ranging from numerical integration to semiconductor device simulation [31],[32].

Monte Carlo applications can be divided into two major groups, the first of which consists of simulation of systems which are statistical in nature. The second group consists of methods designed for problems that have well defined mathematical equations. However, most problems are a combination of the two. For example, transport problems in semiconductors and plasmas are statistical in nature but also have well-defined equations.

Consider an example illustrating the use of Monte Carlo methods to determine the occurrence of a collision for a particle. Let Γ_m be the scattering rate and Δp the probability that the particle will experience a collision during the interval Δt , then

$$\Delta p = \Gamma_m \Delta t \quad (2.20)$$

and the probability of no collision is given by

$$p_{not} = 1 - \Delta p = 1 - \Gamma_m \Delta t. \quad (2.21)$$

This method can be extended to a system containing a number of scattering events λ_i such that $\sum_i \lambda_i = \Gamma$. Consider a set of three collision events whose rates are λ_1 , λ_2 and λ_3 . The random number generated will determine the type of collision undergone by the particle as shown in Figure 2.2. Here the value of a random number, r , lying between 0 and 1 is used to choose one of the four types of collisions. For example, a value of 0.3 will result in a collision event corresponding to λ_2 . The height of each strip in Figure 2.2 is proportional to the relative probabilities of the events. The accuracy of this method depends on the fact that if it is done for a sufficiently large number of particles, the random numbers generated will follow the correct statistical distribution. Figure 2.3 shows a typical electrostatic plasma particle algorithm in which Monte Carlo methods are extensively used. In this algorithm, Monte Carlo determines the type of collision and the state of the particle after collision.

Monte Carlo techniques can also be applied to generate distributions according to a given function. In cases where the distribution of variables or their properties are known or are expected beforehand, it may save simulation

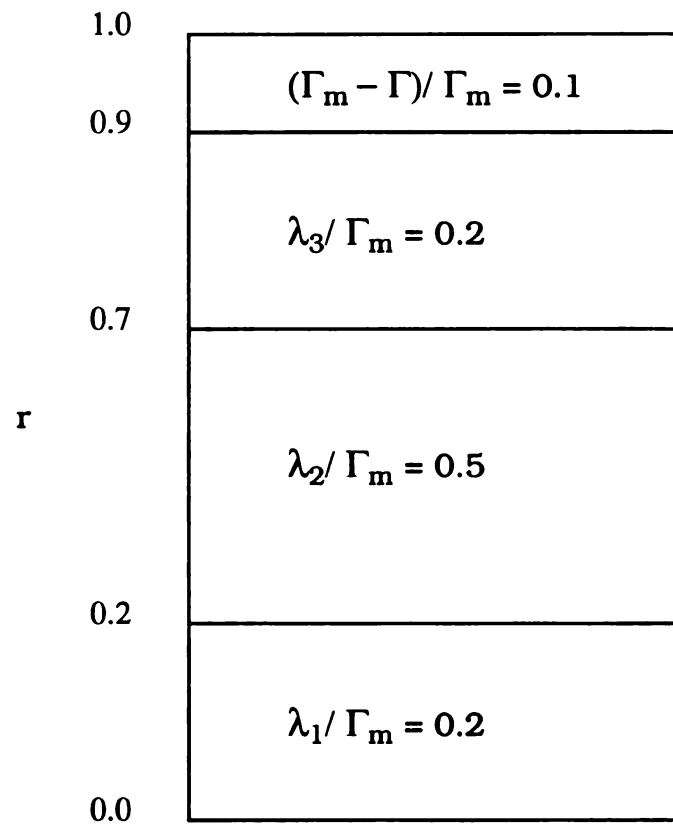


Figure 2.2: Selection of collision event by Monte Carlo method.

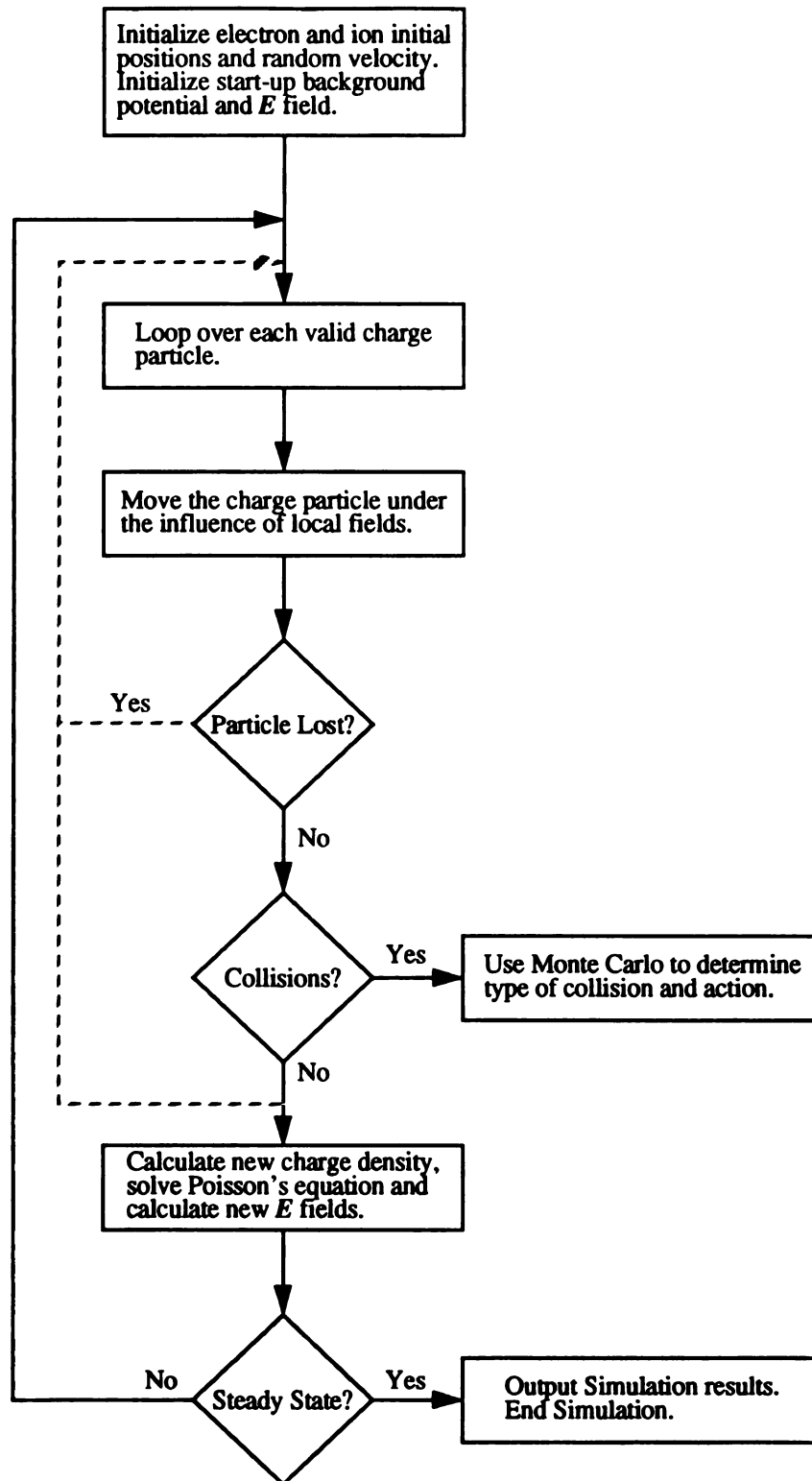


Figure 2.3: Flow chart of a typical electrostatic plasma simulation.

time to initialize the particles according to that function. For example, the start-up velocities of electron and ions can be distributed randomly according to a Maxwell-Boltzmann energy distribution. The same approach can be applied to distribute the random locations in space of the initial set of electrons and ions. In this case, for example, one might follow the ambipolar Bessel function type distribution in space.

The application of the rejection technique to the generation of random numbers [32] with a given distribution $f(x)$ begins by letting C be a positive number such that

$$C \geq f(x) \quad (2.22)$$

in the interval (a,b) , and let r_1 and r'_1 be two random numbers with uniform distributions in the range $(0,1)$. Then

$$x_1 = a + (b-a)r_1 \quad (2.23)$$

and

$$f_1 = r'_1 C \quad (2.24)$$

are two random numbers obtained with a flat distribution in (a,b) and $(0,C)$, respectively. If

$$f_1 \leq f(x_1) \quad (2.25)$$

then x_1 is retained as the choice of x else it is rejected and another pair (r_1, r'_1) is generated and the above process is repeated. The result of the process is a distribution that follows $f(x)$.

Monte Carlo methods have been applied extensively to the study of plasma simulations. B. E Thompson et al. [33] used these methods to simulate ion transport in RF sheaths. In this work, they used a Monte Carlo technique to

select the type of ion-molecule collision (hard sphere, field interaction or charge exchange). Further, the Monte Carlo method was also used to determine the final velocity of ions after elastic collisions. G. R. Govindraju et. al. [34] used Monte Carlo techniques to simulate electron swarm behavior in uniform $\mathbf{E} \times \mathbf{B}$ fields. Here again the method was used to choose one of the many types of collisions. Monte Carlo methods have also been used in plasma etching and sputtering studies. Ignacio et al. [35] applied Monte Carlo techniques to simulate 2D etching of silicon. In this study, angular ion distribution functions were obtained by using Monte Carlo methods to solve the transport of ions across the plasma sheath which were then applied to calculate etch rates. Biersack and Eckstein [36] applied Monte Carlo methods to simulate sputtering yields, energy and angular distributions of sputtered particles in collisional sputtering processes and the simulations agreed with experimental results very closely.

2.2.2 Fluid Models

In higher pressure plasmas (> few mTorr), where the characteristic length of the system is larger than the mean free path of the electrons, i.e., chamber dimensions are several mean free paths long, one can treat the plasma as a fluid. The collective behavior of the plasma components can then be simulated by solving the moments of the Boltzmann equations for each of the species. Fluid model simulations make some simplifying assumptions to terminate the sequence of moments of the Boltzmann equation. For example, a simulation of a one-dimensional plasma with the *cold plasma* approximation for electrons and ions would solve the following equations:

$$\frac{\partial n_e}{\partial t} + \frac{\partial j_e}{\partial x} = k_{ion} n_n n_e - \alpha_r n_i n_e \quad (2.26)$$

$$\frac{\partial n_i}{\partial t} + \frac{\partial j_i}{\partial x} = k_{ion} n_n n_e - \alpha_r n_i n_e \quad (2.27)$$

$$j_e = \mu_e n_e \frac{\partial V}{\partial x} - D_e \frac{\partial n_e}{\partial x} \quad (2.28)$$

$$j_i = -\mu_i n_i \frac{\partial V}{\partial x} - D_i \frac{\partial n_i}{\partial x} \quad (2.29)$$

along with the Poisson's equation in one dimension

$$\frac{\partial^2 V}{\partial x^2} = \frac{q}{\epsilon} (n_i - n_e) \quad (2.30)$$

where k_{ion} is the ionization rate, α_r is the recombination rate, n_n is the neutral gas density, $\mu_{i(e)}$ is the ion (electron) mobility and $D_{i(e)}$ is the ion (electron) diffusion coefficient, $n_{i(e)}$ is the ion (electron) charge density, $j_{i(e)}$ is the ion (electron) current density and V is the potential. It should be noted that of the above set of equations, only equations (2.26), (2.27) and (2.30) are mutually independent since equations (2.28) and (2.29) can be substituted in (2.26) and (2.27) to form a set of three independent differential equations. Since these equations vary in time and space, initial conditions for all three independent variables are specified in addition to boundary conditions which are specified for every timestep at the two boundaries.

Graves [37] simulated plasmas using fluid models where both electrons and ions were treated as continuous fluids. In this model electron and ion motion was assumed to be collisionally dominated. In another example, Barnes et al.

[38] simulated a set of equations similar to (2.26)- (2.30) for low-pressure (50-500 mTorr) RF glow discharges.

2.2.3 Hybrid Models

When electrons and ions are acted upon by the same electric field, electrons, due to their lower mass, traverse a longer distance than the ions in the same timestep. This difference is enhanced in the simulation of low pressure plasmas where each species has a different temperature and hence a different random velocity. For example, in a simple microwave argon plasma at low pressures (0.1-10mTorr), the electron temperature is generally between 4-15 eV while the ion temperature is in the range of 0.3-1 eV. This difference in energies presents difficulties while simulating plasmas. In order to conserve energy and for the simulation to make sound physical sense, the distance travelled by an electron per timestep in particle type simulations should be less than a grid size, which is of the order of the Debye length. However, if ion motion is considered in the same timestep, it would travel less than a thousandth of the grid for the above example. This problem is magnified in magnetically confined plasmas where an electron can typically spend a long time trapped along the magnetic field lines while the ions are unaffected. In order to reduce unnecessary computations, methods have to be developed which de-couple time-steps for electrons and ions and one such method is the fluid-particle hybrid model. Further, Monte Carlo techniques introduced earlier, can turn out to be computationally expensive or difficult when applied to low field regions, slow ion motion and prediction of particle recombination and generation. In these cases, a hybrid model such as

proposed by Bandopadhyay et. al. [39] to couple drift-diffusion models with Monte Carlo methods have been considered.

Hybrid models typically treat one of the particles as a fluid while simulating the other as a particle. For example, a simulation may consider the electrons as a fluid and treat the ions as particles and use a timestep based on ion motion. Electron behavior is simulated by using the moments of the Boltzmann equation similar to the method described in Section 2.2.2.

Lee and Birdsall [40] used hybrid models to simulate beam-plasma interactions. Surendra and Graves [42] used hybrid models to do 1D simulations of DC glow-discharges used in diamond film deposition. In their work, hydrogen discharges were simulated at pressures of 20-30 Torr by treating energetic electrons in the cathode sheath as particles and both electrons and ions in low field region as fluids. The sheath electrons were coupled to the bulk fluid model as a factor in Poisson's equation.

2.2.4 Poisson's and Maxwell's Equations Solution

One way of modeling the electromagnetic excitation of plasma source regions is to directly solve Maxwell's curl equations which govern time dependence of the electromagnetic fields. In addition to these, Poisson's equation is solved numerically to self consistently resolve local static electric fields and charge density. The \mathbf{B} field generated by the permanent magnets can be solved in advance since they do not change for the duration of the simulation [8].

However, the case of downstream behavior simulation is essentially electrostatic, since most of the electromagnetic fields are absorbed by the plasma in the generation region, the electromagnetic fields can be neglected in

the downstream region. In this case, only Poisson's equation needs to be solved and the charged particles are moved under the influence of static electric fields only.

Poisson's Equation can be discretized in rectangular coordinates as follows

$$\begin{aligned} & \frac{(V_{j-1} - 2V_j + V_{j+1})_{k,l}}{\Delta x^2} + \frac{(V_{k-1} - 2V_k + V_{k+1})_{j,l}}{\Delta y^2} + \\ & \frac{(V_{l-1} - 2V_l + V_{l+1})_{j,k}}{\Delta z^2} = -\frac{\rho_{j,k,l}}{\epsilon} \end{aligned} \quad (2.31)$$

where j , k and l are the grid indices and Δx , Δy and Δz are the grid spacing in the x , y and z directions, respectively. Similarly the \mathbf{E} field in the three directions is obtained by solving equation (2.6). For example, E_x , the component of the electric field in the x direction is obtained at grid point (j,k,l) as follows

$$(E_x)_{j,k,l} = \frac{(V_{j-1} - V_{j+1})_{k,l}}{2\Delta x} \quad (2.32)$$

and components in the y and z directions are calculated similarly.

The solutions to Maxwell's curl equations for plasma discharges has been studied extensively by Tan [41] and the reader is referred to his dissertation for details. The method utilized in his study was the finite-difference time-domain (FDTD) method.

Another important part of numerical calculation of plasmas is the particle and force weighting, i.e., the relation between grid and particle quantities. The simplest form of weighting is called zero-order weighting also known as *nearest-grid-point* (NGP) weighting. In this case, all particles in the volume $\Delta x \Delta y \Delta z$ around the grid point (j,k,l) are assigned to that point. This procedure is computationally very fast since only one grid look up is done and charge density

is the number of particles divided by the volume. However, this form of weighting leads to relatively noisy charge density and electric fields.

First order weighting, also called *cloud-in-cell* (CIC) [43] smoothens the above noise but adds the additional computation of looping over the nearest eight grid points in the three dimensional case. In the case of nonuniform grid spacing or grid points on the boundary, the volume around the grid points has to be calculated for each grid point. Consider an internal grid point (j,k,l) forming the origin of a cube of one grid spacing length in each of the three directions, and consider a charged particle located at (x,y,z) such that the nearest grid point is (j,k,l) with location (x_j,y_k,z_l) as shown in Figure 2.4. The x location of the particle inside the cube is $(x-x_j)$, and the y and z locations are calculated similarly. Then the fraction of charge that gets assigned to each of the grid points is done as follows

$$\begin{aligned}
 q_{j,k,l} &= (1-X) \times (1-Y) \times (1-Z) \\
 q_{j+1,k,l} &= (X) \times (1-Y) \times (1-Z) \\
 q_{j,k+1,l} &= (1-X) \times (Y) \times (1-Z) \\
 q_{j,k,l+1} &= (1-X) \times (1-Y) \times (Z) \\
 q_{j+1,k+1,l} &= (X) \times (Y) \times (1-Z) \\
 q_{j+1,k,l+1} &= (X) \times (1-Y) \times (Z) \\
 q_{j,k+1,l+1} &= (1-X) \times (Y) \times (Z) \\
 q_{j+1,k+1,l+1} &= (X) \times (Y) \times (Z)
 \end{aligned} \tag{2.33}$$

where $q_{j,k,l}$ is the contribution of the charge at point (j,k,l) from the particle under consideration, X is the fractional distance of the particle in the x direction from the grid point (j,k,l) and is defined as

$$X = \frac{x-x_j}{\Delta x} \tag{2.34}$$

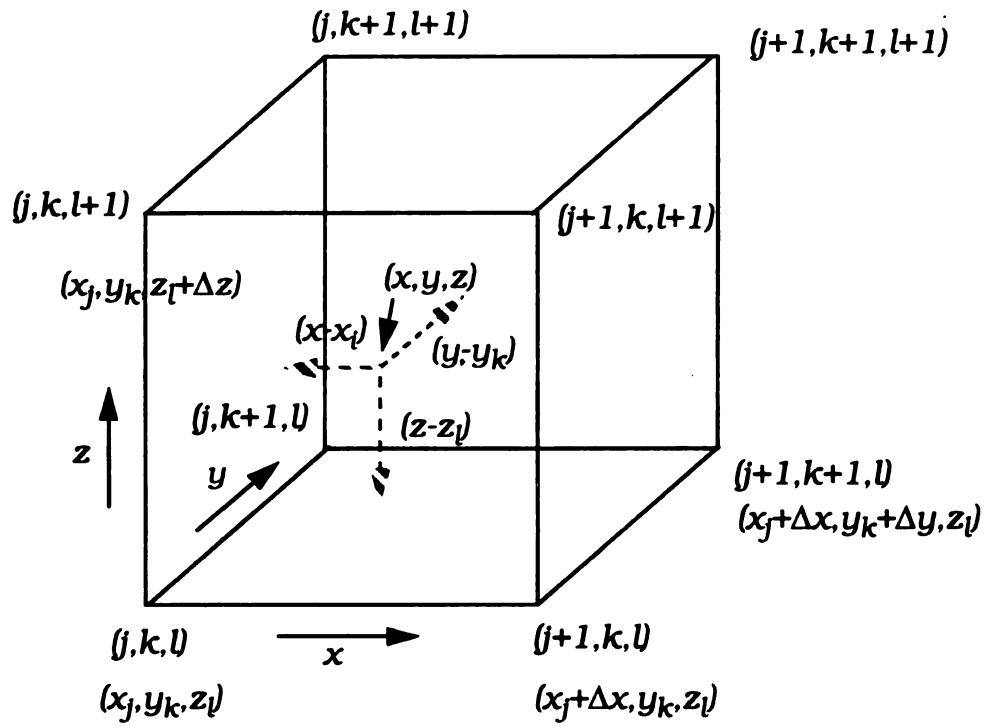


Figure 2.4: Distribution of particle charge to grid.

and y and z are defined similarly. The CIC method has the advantage of weighting the charge on a grid point in accordance to its relative distance from the particle. This form of weighting results in a smoother effective particle shape as compared to NGP. Higher order weighting in the form of quadratic and cubic splines can also be considered at the cost of extra computation.

An inverse form of the above weighting is done while calculating the effective electric and magnetic fields acting on a particle at a fractional distance (X,Y,Z) from the grid point (j,k,l) . In this case the field at the particle is a sum of the weighted field contributions from the eight neighboring grids points. It is important to note here that the same type of weighting should be done for both the charge and the fields in order to avoid a *self-force*, i.e, the particle accelerating itself.

The effect of particle motion appears in the form of current density \mathbf{J} in the right hand side of equation (2.2). Equations (2.1) and (2.2) are then iteratively solved to obtain self-consistent \mathbf{E} and \mathbf{H} fields. Current density due to particle motion is calculated by distributing the particle velocities in each of the three dimensions in a manner similar to equation (2.33). Then, the current density in the x direction at the grid point (j,k,l) , J_x , is calculated as follows

$$J_x(j, k, l) = v_x(j, k, l) \times \rho_{j, k, l} \quad (2.35)$$

where $v_x(j,k,l)$ is the contribution of particle velocities to grid point (j,k,l) and $\rho_{j,k,l}$ is the charge density at that point. Using current density calculated in this manner only approximately satisfies the continuity equation

$$\nabla \cdot \mathbf{J} + \frac{\partial \rho}{\partial t} = 0. \quad (2.36)$$

This could lead to errors in the charge and the divergence of the electric field. One approach is to introduce some corrections to the right hand term [44] in equation (2.31) in order to satisfy the charge continuity equation (2.36). First the electric field \mathbf{E} is modified as follows

$$\mathbf{E}^{n+1} = {}^{(0)}\mathbf{E}^{n+1} - \nabla V \quad (2.37)$$

where \mathbf{E}^{n+1} is the electric field calculated at $(n+1)^{th}$ timestep by solving equations (2.1) and (2.2) and ${}^{(0)}\mathbf{E}^{n+1}$ is the modified electric field used to provide a correction factor to Poisson's equation. The source term in Poisson's equation is then modified as follows

$$\nabla^2 V = \nabla \cdot {}^{(0)}\mathbf{E}^{n+1} - \frac{\rho^{n+1}}{\epsilon}. \quad (2.38)$$

This approach requires explicit solving of Poisson's equation in order to enforce Gauss' law.

Another approach to this numerical problem is to introduce an error correcting term to the current \mathbf{J} [45], [46].

First, a function $F(\mathbf{x}, t)$ is defined as

$$F(\mathbf{x}, t) = \nabla \cdot \mathbf{D} - \rho \quad (2.39)$$

where

$$\mathbf{D} = \epsilon \mathbf{E}. \quad (2.40)$$

The correction term, $d\nabla F$, termed as Marder's correction, added to \mathbf{J} is computed as follows

$$\mathbf{J} = \mathbf{J} - d\nabla F \quad (2.41)$$

where d is a correction factor bound by the constraint

$$d < \frac{\Delta x^2}{2\Delta t} \quad (2.42)$$

and Δx and Δt are the increments in space and time respectively.

2.2.5 Geometrical Considerations

Geometrical considerations play a very important role in plasma simulations. While 2D and 3D rectangular structures can be simulated with simple discretizations, simulating 3D cylindrical structures is more complicated. Firstly, in order to satisfy the Debye length criterion in all dimensions, there will have to be 2π more grid points on the circumference than there are in the radial direction. This could be very expensive in terms of computer memory requirements and advanced computer techniques, discussed in Section 2.4, will have to be considered. Secondly, uniform grid spacing in cylindrical coordinates has another problem. Consider a cylinder of radius R divided uniformly into 10 grid points each of length $R/10$ as shown in Figure 2.5. The ratio of the area between the first annular region, numbered 1 on the figure, and the last region, numbered 10, is $\pi (R/10)^2 / \pi (R^2 - (0.9R)^2)$ and is approximately 1:20, i.e. 5%. If the charge were initially distributed to the grid uniformly, there would be substantially fewer particles in the inner rings than the outer. This will lead to instability of the program at grid points closer to the origin. This problem can be solved in a couple of different ways, one of which is to create nonuniform grids in an attempt to make the outer and inner grid areas the same order of magnitude. In the second method, grids could be created such that there are

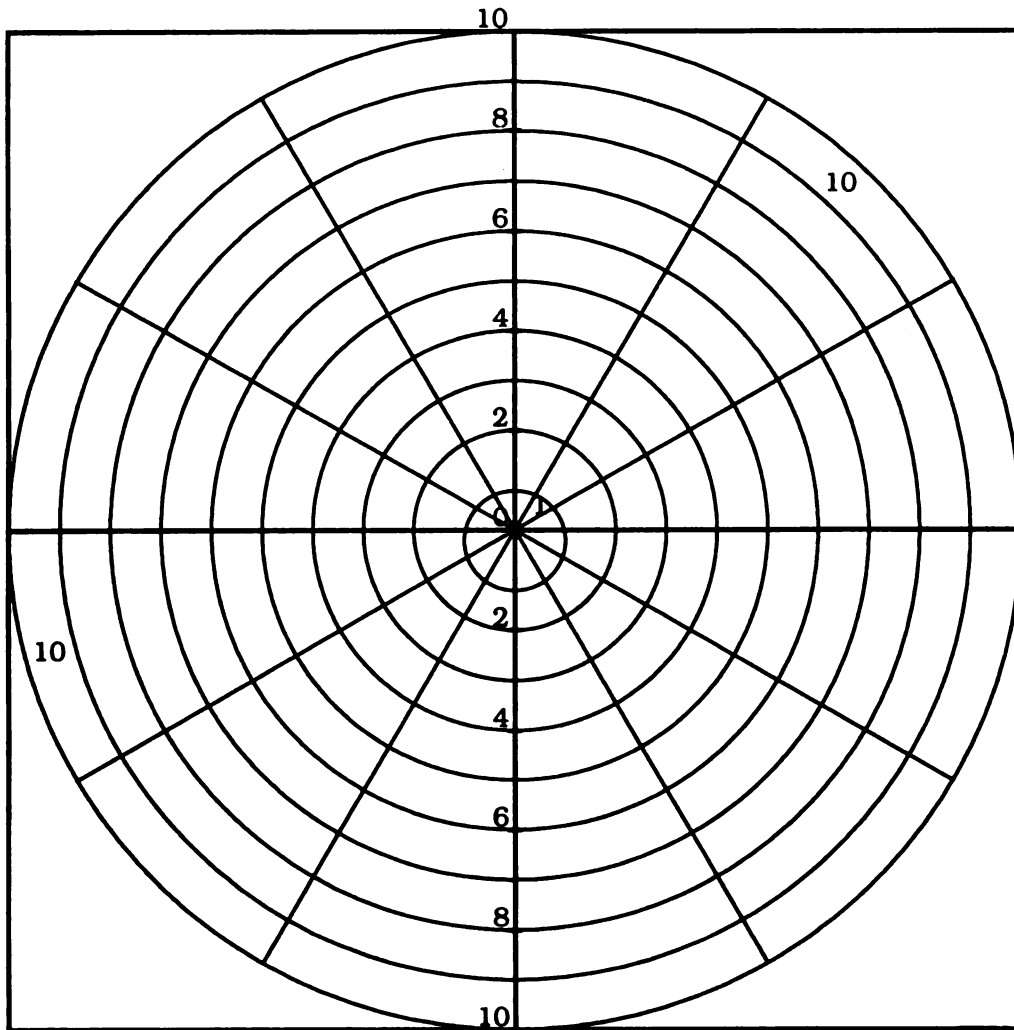


Figure 2.5: Cylindrical grids with uniform grid spacing.

fewer grids in the ϕ direction for radial grids closer to the origin as shown in Figure 2.6. In this case the ratio of the areas of two grid elements is 1:6.4, i.e. 15%. There is still a substantial disparity in areas which indicates that finer ϕ grid spacing is needed for the outer grids. In this case the numerical method used to solve Poisson's equation, for example, Gauss-Seidel method, will have to be modified to reflect the irregularity in one or more dimensions. This structuring has the advantage of reducing the amount of memory used for the structure, however, additional computational time will be required to resolve the nonuniform grid spacing. This approach has been extended and implemented in this dissertation. In either case the objective is to make the ratio of the areas closer to unity.

A combined cartesian-cylindrical coordinate approach can also be used in certain cases. Hese and Zutter [47] simulated a rectangular waveguide with a cylindrical bend in it by discretizing Maxwell's equation in cartesian coordinates for the rectangular part and in cylindrical coordinates for the bend. They developed a transformation relating the values of variables at the boundary with the values in the neighboring grid points in the two regions.

Another approach to simulating regions with curved boundaries is to use non-orthogonal unstructured coordinate systems. In this case the coordinates representing the three dimensions need not necessarily be mutually perpendicular. It should be noted here that conventional vector geometry assumes orthogonal coordinate systems and, in this case, vector equations like the Lorentz force equation will have to be solved with care.

Matsumoto and Kawata [48] have proposed approximating curved boundaries using triangular meshes to solve particle motion in magnetostatic

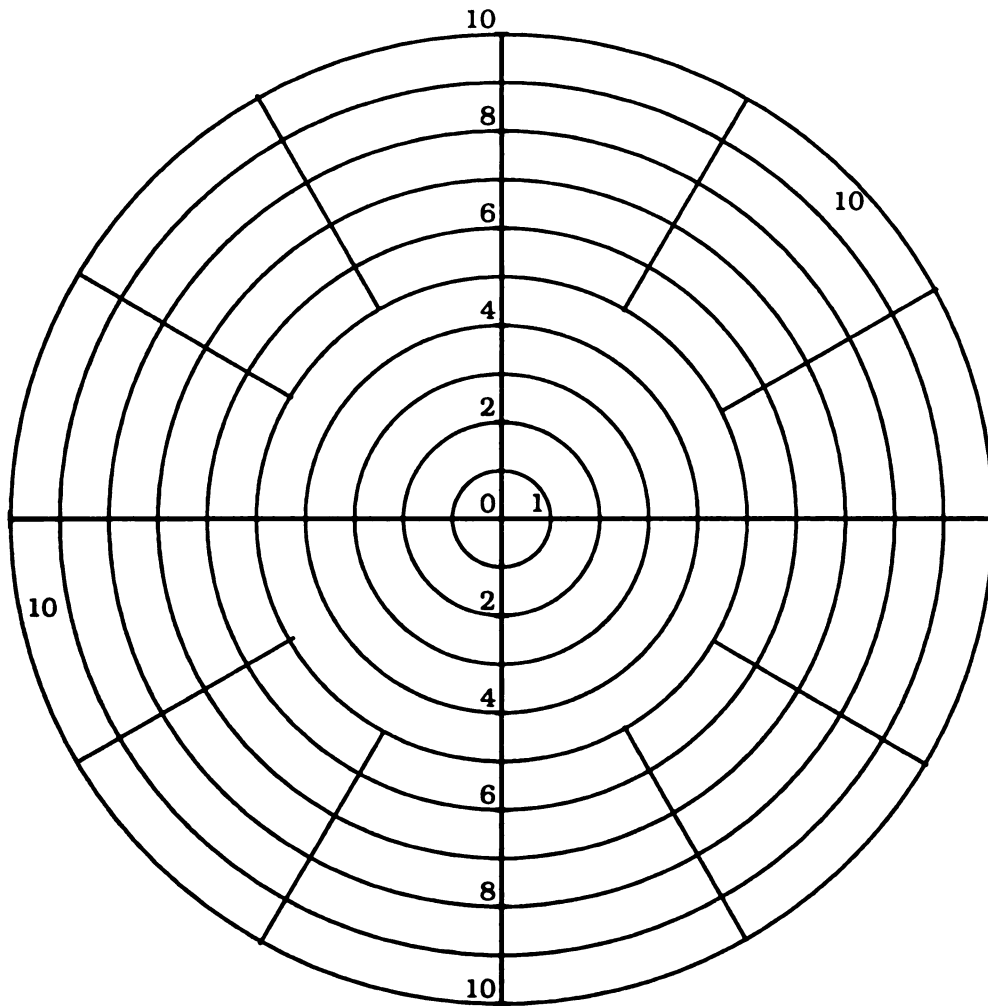


Figure 2.6: Cylindrical grids with non-uniform grid spacing.

fields. Assuming small timesteps they could locate the new position of the particle by searching through a small number of grids. Sonnendruker et al. [49] presented a finite element PIC (particle in cell) model for unstructured meshes. Brandon et al. [18] presented a 3D EM plasma simulation using non-orthogonal unstructured grids. Their algorithm reduced to a standard Finite Difference Time Domain (FDTD) method when orthogonal grids were used.

2.3 Analytical Modeling of Plasmas

In very low density plasmas, electron motion can diffuse through a background neutral gas without any substantial effect due to the ions. This type of diffusion where the coulomb forces can be neglected is termed as *free diffusion*. In higher density plasmas however, the diffusion rate of electrons and ions is controlled by the strong coulomb attraction which tends to maintain overall space charge neutrality. These theories can be derived from the general Boltzmann equation [51] after making several simplifying assumptions. Consider the conservation equation for electrons given by

$$\frac{\partial n_e}{\partial t} + \nabla \cdot (n_e \langle \mathbf{v}_e \rangle) = n_e \nu_i - \alpha n_e^2 - \nu_A n_e - n_e Z_e \quad (2.43)$$

where n_e is the electron charge density, $\langle \mathbf{v}_e \rangle$ is the average value of the electron velocity, ν_i is the ionization frequency, ν_A is the frequency of attaching collisions, α is the recombination co-efficient and Z_e is the generalized source frequency. Similar equations can be written for ions and neutrals. Under steady state conditions and sinusoidal wave forcing fields, each macroscopic quantity can be written as

$$\begin{aligned}\langle \mathbf{v}_e \rangle &= \langle \mathbf{v}_{0e} \rangle + \langle \mathbf{v}_{1e} \rangle e^{j(\omega t - \mathbf{k} \cdot \mathbf{r})} \\ n_e &= n_{0e} + n_{1e} e^{j(\omega t - \mathbf{k} \cdot \mathbf{r})}\end{aligned}\tag{2.44}$$

where $\langle \mathbf{v}_{0e} \rangle$ and $\langle \mathbf{v}_{1e} \rangle$ are the average values of the time invariant and time varying components of the electron velocity respectively and n_{0e} and n_{1e} are the time invariant and time varying components of the electron charge density respectively. Assuming quasi-static approximation where the wavelength $\lambda \rightarrow \infty$ or alternatively $\mathbf{k} \rightarrow 0$ leads to $e^{j(\omega t - \mathbf{k} \cdot \mathbf{r})} \rightarrow e^{j\omega t}$.

The momentum conservation for the electrons can be written as

$$\begin{aligned}\frac{\partial}{\partial t}(n_e m_e \langle \mathbf{v}_e \rangle) + \nabla \cdot (n_e m_e \langle \mathbf{v}_e \mathbf{v}_e \rangle) \\ + n_e q (\mathbf{E} + (\langle \mathbf{v}_e \rangle \times \mathbf{B})) = n_e m_e \langle \mathbf{v}_e \rangle \nu_m\end{aligned}\tag{2.45}$$

where ν_m is the collision frequency for momentum transfer.

Time varying \mathbf{E} and \mathbf{B} fields are also defined in a manner similar to equation (2.44). After collecting only the time invariant terms from equations (2.43) and (2.45) and after considerable simplifications involving the discarding of second order terms, an equation for the collision dominated plasma without volume recombination and attachment can be derived as

$$D_a \nabla^2 n(r, \phi, z) + n \nu_i = 0\tag{2.46}$$

where ν_i is the ionization frequency, n is the plasma density and D_a is the ambipolar diffusion coefficient given by

$$D_a = \frac{\mu_i D_e + \mu_e D_i}{\mu_i + \mu_e}\tag{2.47}$$

where $\mu_{l(e)}$ is the ion (electron) mobility and $D_{l(e)}$ is the ion (electron) diffusion coefficient.

Some of the analytical diffusion theories which are derived after simplifying assumptions to equation (2.46) are explained in the following subsections.

2.3.1 Schottky Diffusion Model

A simple model of diffusion in plasmas is the Schottky diffusion model proposed by Schottky [50] which assumes that volume collisions and space-charge effects govern charged particle diffusion and control the discharge density distribution. This model is used to solve the general diffusion equation (2.46) which is the continuity equation assuming no externally applied electric fields and a *cold plasma* model. Assuming no static \mathbf{B} field, single stage ionizations only and perfectly absorbing boundaries, equation (2.46) can be solved for a finite-length cylinder using a separation of variables technique to give

$$n(r, z) = N_0 J_0 \left(\frac{2.405}{R} r \right) \cos \left(\frac{\pi z}{L} \right) \quad (2.48)$$

where N_0 is the charge density at $z=0$ and $r=0$ and J_0 is the zeroth order Bessel function. The equation relating geometrical variables to the diffusion constant and ionization frequency is

$$\frac{\nu_i}{D_a} = \left(\frac{1}{\Lambda} \right)^2 = \left(\frac{2.405}{R} \right)^2 + \left(\frac{\pi}{L} \right)^2 \quad (2.49)$$

where R is the radius and L is the length of a ϕ symmetric cylindrical discharge chamber. The separation constant, Λ , is defined as the characteristic diffusion length for Schottky diffusion. In low pressure plasmas, where electron

temperatures are much higher than ion temperatures, equation (2.47) can be simplified to a modified form of the diffusion coefficient as

$$D_a = \frac{\mu_i D_e}{\mu_e} = \frac{\mu_i k_b T_e}{q} \quad (2.50)$$

where T_e is the electron temperature, k_b , is the Boltzmann constant and q is the electron charge. Using equation (2.50) and after extensive derivations [23] equation (2.49) can be expressed in the form

$$(CP\Lambda)^2 = \frac{(k_b T_e)^{5/2}}{\int_{\epsilon_i}^{\infty} \epsilon \sigma_i(\epsilon) \exp\left(-\frac{\epsilon}{k_b T_e}\right) d\epsilon} \quad (2.51)$$

where P is the pressure of the neutral gas, ϵ is the energy, $\sigma_i(\epsilon)$ is the ionization cross section, ϵ_i is the ionization potential and C is a constant expressed in units of $(\text{Torr-cm})^{-1}$ when the pressure is expressed in Torr and the length in cm. It should be noted that in order to arrive at equation (2.51), it was assumed that ion mobility varies inversely with pressure. Equation (2.51) provides a simple relationship between geometrical factors, represented by Λ , neutral gas pressure P and electron temperature T_e and hence can be used to predict scaling relations in low pressure cylinder type plasmas.

2.3.2 Free-Fall Diffusion Model

Another analytical model of low pressure, uniform discharges is the Free-Fall diffusion theory. This theory assumes that the mean free path is greater than the cavity dimensions and that the transport equations are dominated by ionization collision terms. Ions and electrons are assumed to freely fall to the discharge walls and quasi-neutrality is assumed everywhere except in the

sheath. This theory assumes that the only loss mechanism of ions and electrons is due to the recombination at the walls and hence derives a particle balance equation between loss to the walls and the gain due to ionization. Assuming uniform ion flux and charged particle density, the balance equation can be written for a cylindrical structure as described in Section 2.3.1 as

$$\begin{aligned}\Gamma_i A_{wall} &= n_i v_i V_{dis} \\ \Gamma_i 2\pi (RL + R^2) &= n_{io} v_i \pi R^2 L\end{aligned}\tag{2.52}$$

where A_{wall} is the surface area of the wall, Γ_i is the ion flux to the walls, V_{dis} is the chamber volume and n_{io} is the constant ion density in the chamber. A general diffusion expression for free-fall diffusion can be found [51] as

$$CP\alpha = \frac{(k_b T_e)^2}{\int_{\epsilon_i}^{\infty} \epsilon \sigma_i(\epsilon) \exp\left(-\frac{\epsilon}{k_b T_e}\right) d\epsilon}\tag{2.53}$$

where C is a constant expressed in $1/(\text{cm}^3 \text{Torr})$ when the pressure P is expressed in Torr, σ_i is the ionization collision cross section expressed in cm^2 , the length terms are expressed in cm and

$$\alpha = \left(\frac{RL}{R+L} \right).\tag{2.54}$$

Mahoney [23] compared experimental measurements on an ECR plasma reactor with both the free-fall and Schottky diffusion theories and showed that for his reactor which was 8.9cm in discharge diameter, the plasma characteristic makes a transition from Schottky type behavior to a free-fall type behavior below a pressure-diffusion length of 0.015 Torr-cm and an electron temperature of 36,000 K. He explained this was due to the fact that at smaller volumes and

lower pressure, the mean free path of ions exceeds chamber dimensions and charge density can be considered essentially uniform, i.e., the assumptions of the free-fall diffusion are more valid in this region.

2.3.3 Ambipolar Diffusion Model

Experimental plasma reactors of the type shown in Figure 2.7 are not of a simple cylindrical form assumed in the earlier sections. In fact, they consist of a source region which expands into a larger downstream processing region and equation (2.48) no longer accurately predicts the variation in charge density within the chamber. An extension of the Schottky diffusion model to a more complex geometry was done by Hopwood et. al. [3]. This work used an ambipolar diffusion model to describe the downstream behavior of the plasma. It was assumed that the electrons and ions follow ambipolar diffusion in which a space charge region slows the electron motion and speeds up the ions such that both move at the same diffusion rate given by equation (2.47). This theory also assumes equal charged particle densities and fluxes in the plasma.

Solving the ambipolar diffusion equation (2.46) for the processing region with a uniform source discharge radius b and a chamber radius a as shown in Figure 2.7 gives an analytic expression for ion density as [3]

$$n(r, z) = N_0 \left(\frac{2b}{a} \right) \sum_{(n=1)}^{\infty} \left(\frac{1}{x_n} \right) \frac{J_1 \left((x_n) \frac{b}{a} \right)}{J_1^2(x_n)} J_0 \left((x_n) \frac{r}{a} \right) \exp(-kz) \quad (2.55)$$

where

$$k = \sqrt{\left(\frac{x_n}{a} \right)^2 - \frac{v_i}{D_a}} \quad (2.56)$$

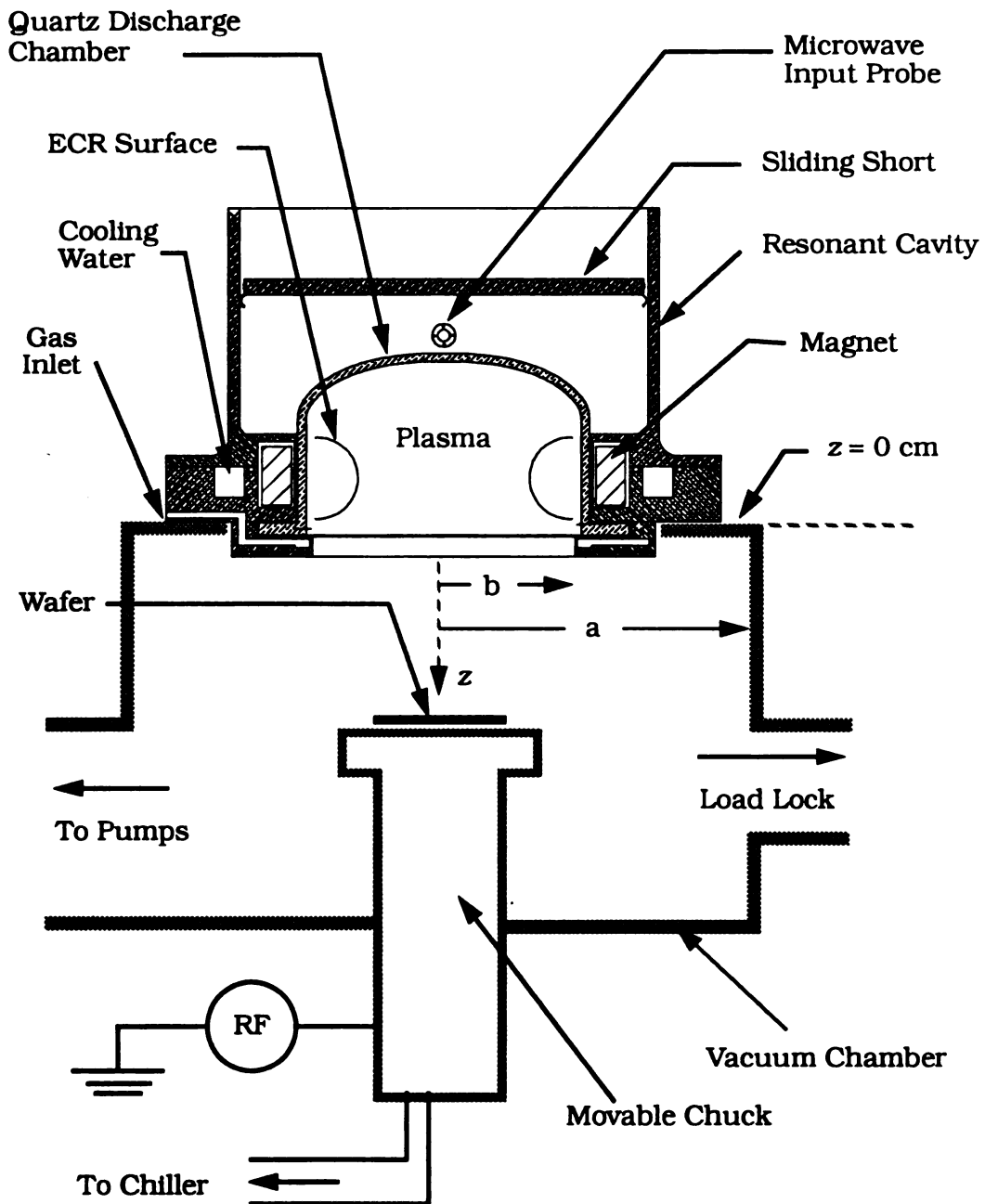


Figure 2.7: A 25 cm diameter ECR system.

and N_0 is the charge density at $z=0$ and $r=0$, J_n is the n th order Bessel function of the first kind and x_n is the n th zero of J_0 . In solving the equation angular symmetry is assumed, hence there is no ϕ dependence. As observed in equation (2.55), the density distribution has a Bessel function dependence on the radial variable, r , and a decaying exponential dependence on the vertical variable, z .

Equation (2.55) has shown good agreement in some studies with experimental data. In one such study, the sputtering uniformity versus position on a wafer was used as a measure of the ion density variation. This was done by Salbert [4] on a 9 cm system by measuring both the ion density and the sputtering rate versus radial position at a position 8 cm downstream from the plasma source. The density of argon ions was measured using a double Langmuir probe and it was seen that the plasma density followed the ambipolar diffusion model closely, and the amount of oxide sputtered away was proportional to the ion density. Hence plasma density given by equation (2.55) can be used to predict the uniformity and relative sputter rate of argon discharges. Equation (2.55) indicates that the uniformity of the plasma density and hence the sputtering rate across a given diameter is improved by using a larger diameter plasma source.

2.3.4 Limits of Analytical Models

The analytical model derived in the above section assumed a constant source region ion density and thus fails to accurately predict a downstream discharge in the case of a nonuniform source. This theory was also derived under the assumption of ϕ symmetry and hence would have to be re-derived for cases where ϕ symmetry is not present. Furthermore, this theory needs to be modified

for the case of downstream magnetic confinement where the condition of absorbing walls is no longer true.

The application of simple analytical models is helpful in plasma reactor scaling. More accurate models which would reduce the design and optimization time for plasma sources and processes will need to utilize numerical techniques as described in the earlier sections of this chapter.

2.4 Vector, Parallel and Distributed Processing Methods

Plasma simulations, whether they be purely particle type or hybrid, typically solve Poisson's equation in order to self consistently resolve charge density and electric field. This requires discretization of Poisson's equation over the three dimensional grid points. The charge and current density calculations have to be done using the particle-in-cell (PIC) approach to distribute the effect of the charged particles over the entire grid structure. Hence, the effect of larger geometries is twofold as it increases the demand on the computer's memory and the time taken per iteration due to the larger grid size and increased number of particles.

The effects of the above problems can be reduced by using innovative computer techniques. Depending on the symmetry of the structure under simulation, one can accurately predict the whole structure by simulating a slice of the same. For example, in an ECR system with four magnets providing the ECR fields, one can predict the whole structure by simulating only 90 degrees and then reflecting the results around the circle as shown in Figure 2.8. This approach can save up to three-quarters of the grid points required for

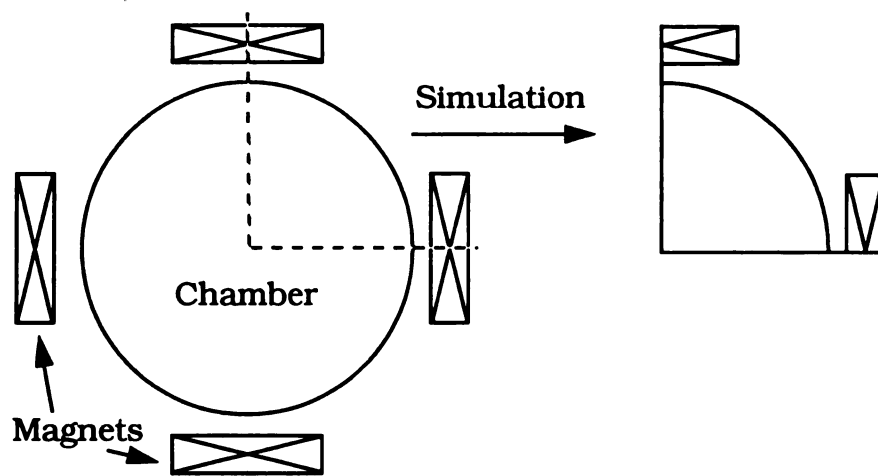


Figure 2.8: Example of symmetry in a plasma simulation.

simulations. This will also reduce the number of particles required to conduct an accurate simulation.

Another approach to this problem is to take the supercomputer route which uses large memory and high-speed resources. This method has its advantages when combined with the “slice” approach and special programming techniques which are unique to the choice of the supercomputer. Lohner and Ambrosiano [21] have developed vectorized particle location tracers for application in vector computers. They reduced the search time for particles by using the last known position as the initial guess and looping over adjacent grids to locate the current position. This allowed them to form vector loops and their results showed a speedup of 14 times when used on vector machines.

The third approach is to use a massively parallel processor (MPP) and distribute the processing across the nodes (CPUs) of the machine. MPPs are designed to reduce interprocessor communication overhead. This approach is extremely desirable in cases where the application is inherently parallel such as in the particle simulation of plasmas. In such simulations, each processing element can be considered to represent one or more grid points and an incremental volume around these grid points is assigned to it. In an extreme case, each superparticle being simulated can be considered a processor. However, a trade-off between the processing speed-up and communication overhead between the processors exists. Lin et. al. [20] have developed parallel PIC codes for a massively parallel processor.

A variation of the above approach would be to use many slower computers in parallel to speed up the program. For example, in the case of a cylindrical cavity, one can conceive of a situation where the cavity is broken up into eight pieces,

each slice of 90 degrees and half of the original z length. Each of these pieces could then be “distributed” to eight different workstations of similar specifications to be simulated. Any data that has to be transferred, for example, a particle leaving one slice and entering another, would require that the details of the particle be transferred from one machine to the appropriate one. This and the potential on the faces “shared” by two machines can be transmitted between the appropriate machines at the end of each “loop” so that all machines are consistent and in synchronization. This approach has the advantage of using cheaper and more widely available workstations to perform simulations. Further, one can conceivably simulate large asymmetrical structures using numerous workstations. This approach also distributes the memory requirement across the participating computers.

A message passing package which caters to such applications is Parallel Virtual Machine (PVM) [52]. PVM has been applied extensively to molecular dynamics problems [53], [54] using Lennard-Jones type potentials to calculate force and transmitting data of any molecules that transfer across. This problem is simpler than plasma discharge simulations since it does not involve calculating any kind of Poisson’s equation which requires more data to be shared between processors. However, the parallel approach has its pitfalls, one of which is stated in the famous “Amdahl’s law” which states that no matter what one tries, there will always be some non-parallel part of the code which will partially offset any gains made by the parallel part. Also the program may spend more time in the communication part thus offsetting any gains due to parallelization.

Use of parallel processing methods generally lead to more complicated code which tend to be very difficult to debug. Therefore new code development and debugging tools, for example *xab* [55], which are specifically developed for parallel programming, will have to be used. The distribution of code across several machines makes it more difficult to pinpoint the problem and would require complex intermediate data storage to make them reliable and debuggable.

Ewing et al. [56] have used PVM to successfully distribute the computation of elastic wave propagation to simulate a displacement seismogram. This implementation was done using a host/node approach where the host program performs I/O and decides the domain decomposition of the nodes. The communication of interprocessor boundary solutions synchronizes the nodes. They demonstrated a speedup of nearly 5.4 for a six processor implementation.

Liewer et al. [57] have studied concurrent PIC codes for plasma simulations and Brandon et al. [18] have studied PIC models for multiprocessors. Matsushita et al. [22] have applied parallel techniques to the simulation of nonlinear MHD (Magneto-Hydro Dynamic) simulations.

2.5 Review of Previous ECR Plasma Modeling Work

Grotjohn [8] did a 2D3V simulation of a compact ECR source assuming axial symmetry. The time varying electric and magnetic fields were solved by using time-domain finite-difference solutions of Maxwell's equation. The electron and ion behaviors were modeled using Monte Carlo particle techniques.

Kuckes [58] computed the propagation and absorption of electromagnetic waves in magnetized plasmas. He predicted complete wave absorption above a

critical plasma density. Trost and Shohet [59] simulated electron behavior during ECR heating in a stellarator. They used Monte Carlo methods to simulate electron distribution functions. The simulation results agreed well with experiments. Sprott [60] developed analytical models to predict ECR heating in toroidal octupoles. A computer simulated static \mathbf{B} field of the octupole was used in the study. His model predicted strong heating in places where $\nabla_{\parallel}\mathbf{B} = 0$. Experimental measurements verified the predicted heating locations. Lieberman and Lichtenberg [61] treated the theory of ECR heating in a magnetic mirror both analytically and numerically. Hussein and Emmert [62] in their simulation of the ECR source region eliminated the need to solve Poisson's equation by assuming quasi-neutrality and inferring the local plasma potential from the electron Boltzmann equation. It has been shown by Self and Ewald [63] that the hybrid approach can be applied even in the case of low pressure free-fall conditions.

Porteus and Graves [64] simulated two dimensional low pressure magnetically confined plasmas by using the hybrid approach. They treated the electrons as a fluid and the ions as particles and Poisson's equation was explicitly solved to obtain the potential. Their simulation of ECR heating assumed either a flat power per electron or a Bessel function type of spatial power deposition. They did not couple Maxwell's equation to self consistently simulate energy.

In their more recent work, Hussein and Emmert [65] studied the ion energy spectrum in divergent static magnetic fields. Ions were treated as particles and ion-neutral collisions were treated using Monte Carlo techniques.

Weng and Kushner [66] investigated the electron energy distribution in the ECR zone using Monte Carlo simulations and explicitly considered electron-electron collisions. The transfer of energy from the electron gas to the background gas for downstream processing was also investigated.

Graves, Wu and Porteus [67] developed 2D hybrid simulations of high density ECR plasmas. The hybrid mode simulation treated electrons as a fluid and ions as particles. The ECR reactor being simulated was not distinguished into a source and downstream region, instead, the substrate was located in the source region itself. Power deposition profiles were assumed such that the ECR zone was near the substrate surface. The effect of radial microwave power deposition profile on plasma parameters like density, potential and electron temperature was investigated.

Yasaka et al. [68] have done 2D modeling of ECR plasma production assuming the cold plasma theory in which the thermal motions of charge particles was neglected. Their simulations developed profiles of EM wave and plasma density evolution in a bounded cylindrical system. Further, they showed that the location of the microwave antenna had considerable effect on plasma production.

Chapter 3

Description of Microwave Plasma Sources

3.1 Introduction

The operating characteristics and description of the two sources used in this research are described in this chapter. A 25 cm diameter MPDR 325 system and a 9 cm diameter plasma discharge system were used for performing some downstream measurements and simulations. A smaller MPDR 610 compact ion source was used for plasma source simulations. This chapter also applies the ambipolar diffusion model of Section 2.3.3 to the problem of modeling the downstream plasma behavior in MPDR 325. An evaluation of the accuracy and limitations of this model are presented and the need for numerical models is discussed.

3.1.1 Downstream Plasma Chamber

The microwave plasma source [3] shown in Figure 2.7 consists of a cylindrical cavity which can be adjusted to a resonant mode by means of the variable length microwave input probe (antenna) and the adjustable sliding short. The lower termination of the microwave plasma source, referred to as the baseplate, is made of several individual parts. A circular quartz discharge chamber, which serves as the containment for the plasma, is surrounded by several rare earth magnets. These magnets are imbedded in the base plate with

alternating poles facing the discharge. Process gas flows into the chamber from holes just under the quartz dome. Inside the vacuum chamber is the anodized aluminum substrate holder that supports the silicon wafer. The RF biased substrate chuck is movable, allowing adjustment of the wafer position with respect to the plasma source region. Experiments were conducted on two ECR systems with 9 cm and 25 cm diameter discharges.

Silicon forms a 4-10 Angstroms thick native oxide within minutes of exposure to air, therefore an in situ, low damage, method to remove small amounts of oxide and other contaminants prior to metallization and other processing steps is desirable. One way to achieve this is by a "soft sputter etch" of the surface using an argon plasma in which Ar^+ ions are accelerated to the surface with low kinetic energy. In some cases, this procedure can be done in the same processing chamber used for subsequent deposition or growth of desired layers.

A prior investigation utilizing an electron cyclotron resonance (ECR) argon plasma excited by 2.45 GHz microwave power for sputter etching was reported by Salimian et. al. [69]. In their divergent field ECR system, which used two electromagnets, the substrate was biased with 13.56 MHz RF power inducing a negative substrate bias. This allowed separate control of plasma density and ion energy as opposed to a conventional parallel plate RF plasma system in which it is difficult to achieve high densities without high ion energies. Salimian et. al. investigated the variation in sputter rate versus microwave power, bias voltage and downstream distance. Potential problems associated with sputter cleaning such as substrate backscattering and redeposition have been reviewed by Vossen [70]. Recent work by Raynaud and Pomot [71] on the cleaning of silicon

surfaces by argon ECR plasmas with low ion energies (< 100 eV) at a pressure of 0.5 mTorr has shown it is feasible to obtain a silicon surface exempt from carbon and oxygen contaminants.

In this chapter experimental sputter oxide removal is investigated by studying both the sputter rate and the uniformity of SiO_2 removal from oxidized silicon wafers using a multipolar ECR argon discharge. Additionally, this chapter correlates the sputter results to plasma measurements. Two permanent magnet based ECR systems with different diameter discharges were used. The oxide thickness versus position was measured with an ellipsometer both before and after the sputter removal process in order to assess uniformity and sputtering rate of 3 and 6 inch wafers. The sputtering rate and uniformity was studied across a parameter space consisting of variations in microwave power, pressure, induced substrate bias and substrate position.

The oxide removal results are correlated with plasma properties including ion densities measured with Langmuir probes. The uniformity results are compared with the ambipolar diffusion model proposed by Hopwood [3]. This analytical ambipolar model introduced in Section 2.3.3 provides a simple method of predicting charge density variation in the radial and axial directions. This model is applied to the prediction of sputter etch uniformity using an argon plasma with the underlying assumption that sputter rate and uniformity are dictated by the plasma charge density at the surface. The amount and uniformity of SiO_2 etched is then compared with the ambipolar model.

3.1.2 Experimental Methods

3.1.2.1 Sample Preparation and Processing

Thermally oxidized silicon wafers with oxide thickness of 500 - 700 Angstroms were scanned by a Gaertner wafer-scanning ellipsometer, providing a map of refractive index and oxide thickness across the wafer surface before and after sputtering. In the case of the 9 cm diameter discharge system ([4],[5],[8]), sputtering was done at a pressure of 1.0 mTorr, a microwave power of 200 W, an argon flow of 7.5 sccm, and a DC bias of -50 V with the wafer located 8 cm downstream from the bottom of the base plate. Sputtering runs were conducted on the 25 cm diameter system by varying plasma conditions over the parameter space of input power, chamber pressure and RF bias voltage. The nominal conditions were 20 sccm argon flow, 1 mTorr pressure, -50 V induced DC bias and 500 W microwave power. Runs were conducted from 0.6 mTorr to 2 mTorr pressure, from -30 V to -80 V RF induced DC bias, from 200 W to 800 W microwave power and with z locations of 6.3 cm, 8.3 cm, 9.8 cm and 11.8 cm downstream.

3.1.3 Experimental Results of Uniformity and Sputter Rate

The sputtering uniformity versus position on the wafer is primarily determined by the incident ion density and energy across the wafer. For the results reported here, the uniformity is determined by the ion density variation. This was verified on the 9 cm system by Salbert [4] by measuring both the ion density and the sputtering rate versus radial position at a position 8 cm downstream from the baseplate. The density of argon ions was measured using

a double Langmuir probe. This result is repeated in Figure 3.1 for continuity with the rest of this chapter. Figure 3.1 shows both the ion density and the oxide removed for a 3 inch diameter wafer. The discharge conditions were -50 V induced DC bias, 1 mTorr pressure and 200 W microwave power. Also shown in Figure 3.1 is the plasma density versus radial position as calculated using the ambipolar diffusion model, with $a = 9.8$ cm and $b = 3.75$ cm (see Figure 2.7), given in Section 2.3.3. It is noted that a and b approximately represent the sizes in the 9 cm system, i.e., their values are set to give the best fit to experimental data. As can be seen in this figure, the plasma density follows closely the ambipolar diffusion model, and the amount of oxide sputtered away is proportional to the ion density. Hence plasma density given by equation (2.55) can be used to predict the uniformity and relative sputter rate of argon discharges. For purposes of comparison, in this report, uniformity is expressed as a standard deviation percentage, which is the standard deviation of the oxide removed divided by the average amount of oxide removed. For the data shown in Figure 3.1, the uniformity was 9%. Equation (2.55) indicates that the uniformity of the plasma density and hence the sputtering rate across a given diameter is improved by using a larger diameter plasma source.

In the case of the larger 25 cm diameter discharge system, the results for the 3 inch diameter wafers versus variation in microwave power, pressure, RF induced bias and downstream position are shown in Tables 3.1, 3.2, 3.3 and 3.4, respectively. Figures 3.2, 3.3, 3.4 and 3.5 show the plots of sputtering rate vs. microwave power, pressure, RF induced DC bias and downstream distance respectively. In all cases the standard deviation percentage uniformity for 3 inch wafers was 2-4%. Of particular note is the improved uniformity with increase in

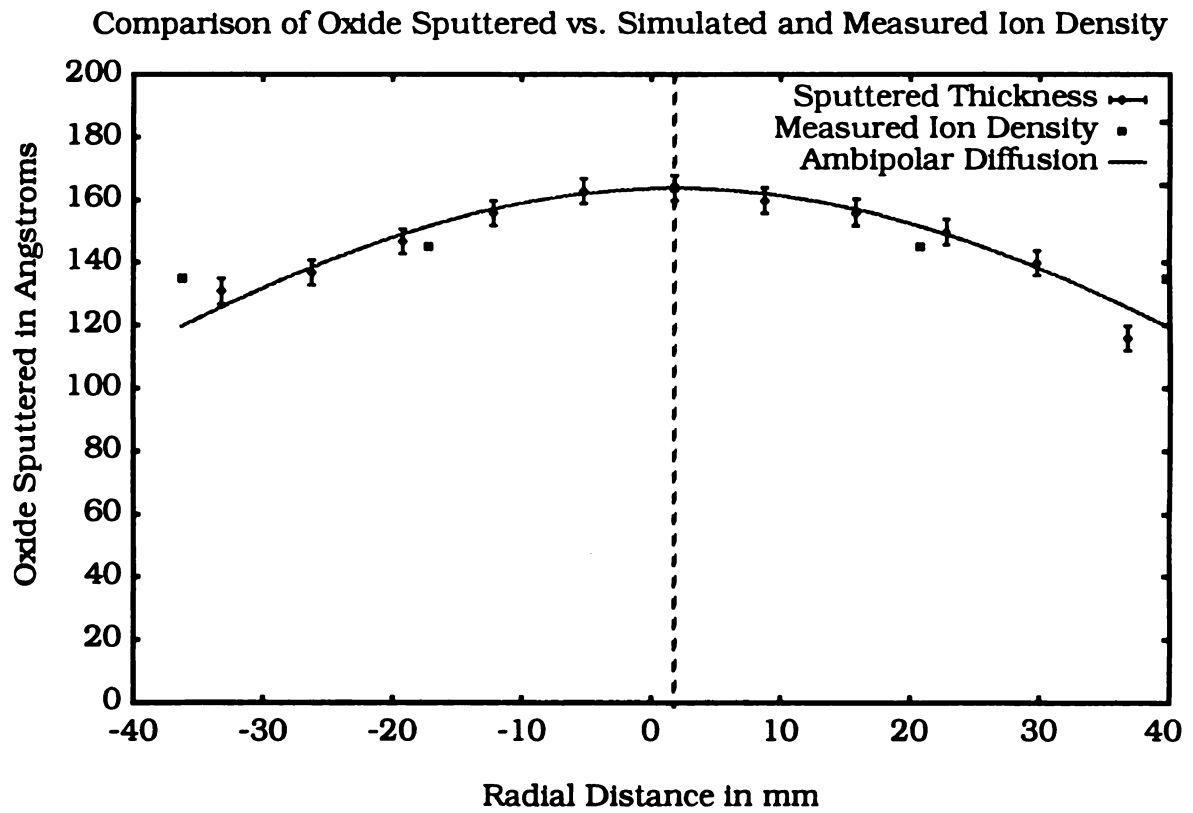


Figure 3.1: Comparison of theory with experiment for 9 cm system.

Table 3.1: Sputtering rate and uniformity vs. microwave power

Power Watts	Sputtering Rate Angstroms/min	Uniformity %
200	20.78	4.00
500	42.64	3.2
800	50.3	3.19

Table 3.2: Sputtering rate and uniformity vs. pressure

Pressure mTorr	Sputtering rate Angstroms/min	Uniformity %
0.6	39.15	2.8
1.0	31.76	3.46
2	23.15	3.6

Table 3.3: Sputtering rate and uniformity vs. RF induced bias

RF induced bias in Volts	Sputtering Rate Angstroms/min	Uniformity %
-30	16.42	3.89
-40	31.76	3.46
-50	42.64	3.2
-65	54.14	3.21
-80	79.22	3.36

Table 3.4: Sputtering rate and uniformity vs. downstream distance

Downstream distance in cms	Sputtering Rate Angstroms/min	Uniformity %
6.3	56.5	2.87
8.3	47.7	2.36
9.8	41.35	2.12
11.8	39.93	2.12

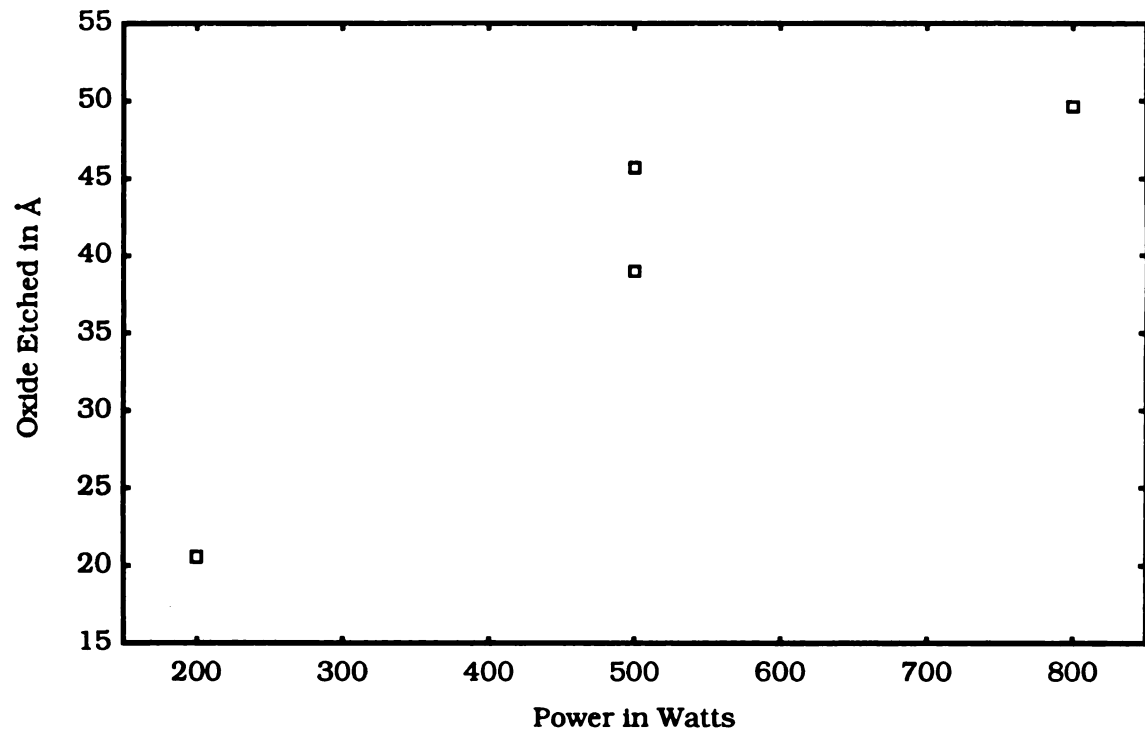


Figure 3.2: Oxide etched vs. power.

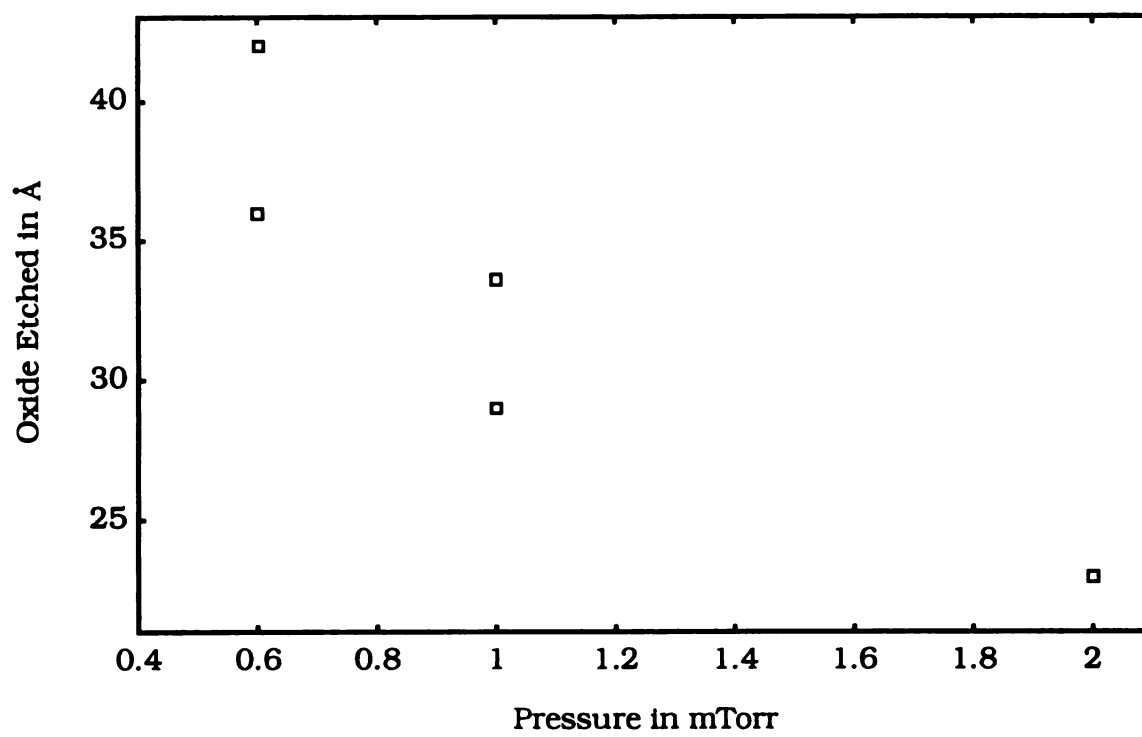


Figure 3.3: Oxide etched vs. pressure.

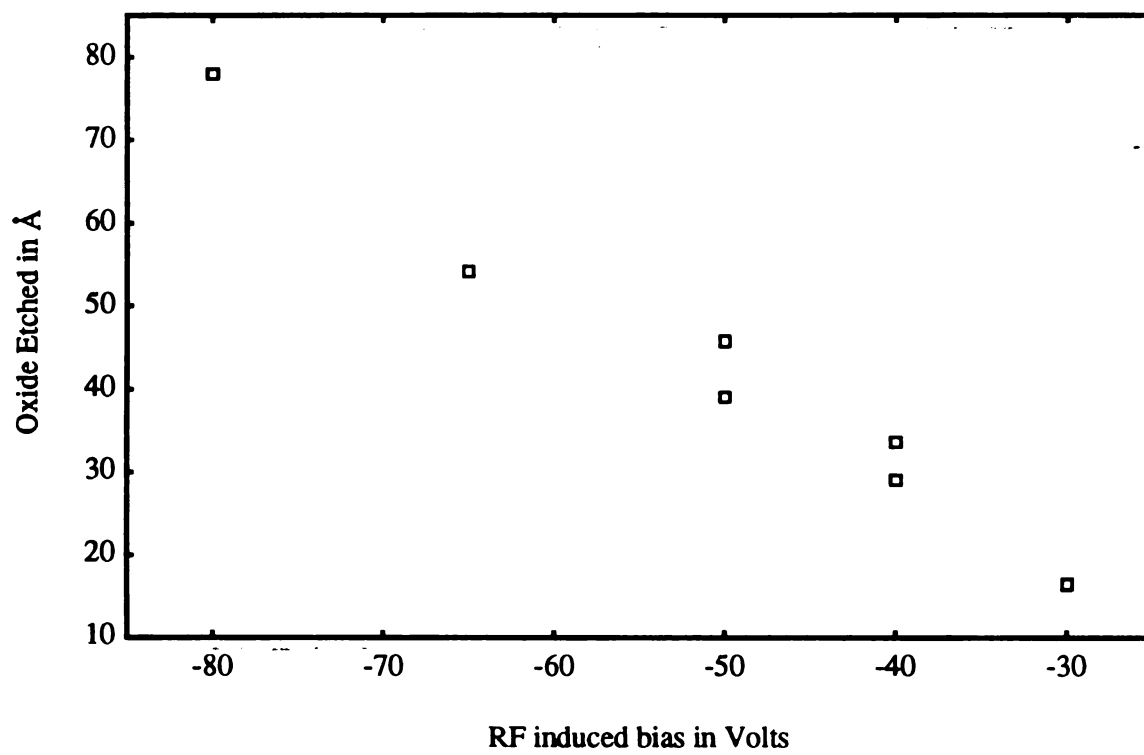


Figure 3.4: Oxide etched vs. RF induced bias.

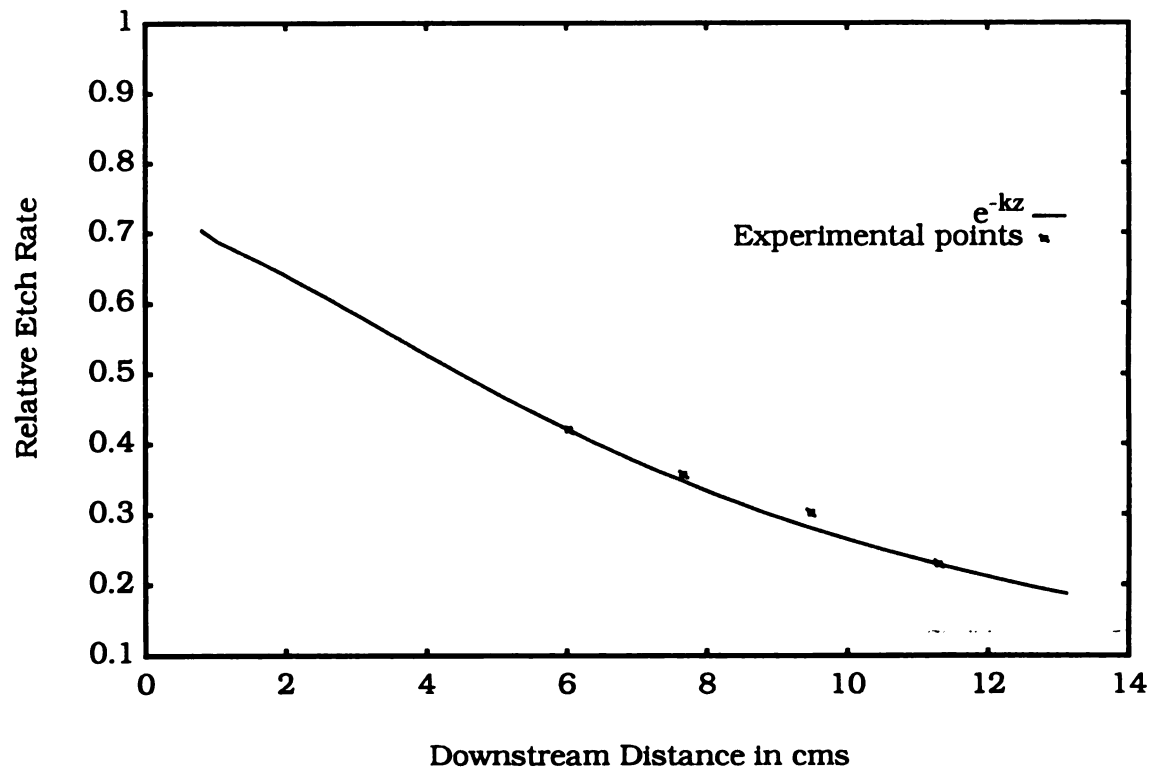


Figure 3.5: Comparison of theory with experiment for downstream points.

downstream distance. The improved uniformity is accompanied by a reduced sputtering rate further downstream. The sputtering rate decrease for position further downstream is predicted approximately by the plasma density given by equation (2.55) and as shown in Figure 3.5. The uniformity also improves as pressure is reduced as given in Table 3.2. This is as expected since the lowering of pressure increases the mean free path between collisions resulting in improved uniformity.

Table 3.1 shows that the sputtering rate increases with microwave power. This occurs because the plasma density increases as the microwave power is increased [72]. Lastly, the sputtering rate decreases as the pressure is increased. This occurs because the plasma potential decreases as the pressure increases [73]. In particular it should be noted that the energy of the ions hitting the surface is approximately given by the sum of the plasma potential and the induced DC bias.

The comparison of sputtering uniformity vs. radial distance with the ambipolar model given in Equation (2.55) is shown in Figure 3.6 for 3 inch wafers. It can be seen that the sputter rates follow the ambipolar predictions closely both in the radial and downstream directions. For the 6 inch diameter wafers the variation in uniformity across the wafer versus radial distance is shown in Figure 3.7. The most oxide was removed in the center with smaller amounts removed near the edge as predicted by equation (2.55). The oxide removal pattern is ϕ symmetric and the uniformity was approximately 5.4%.

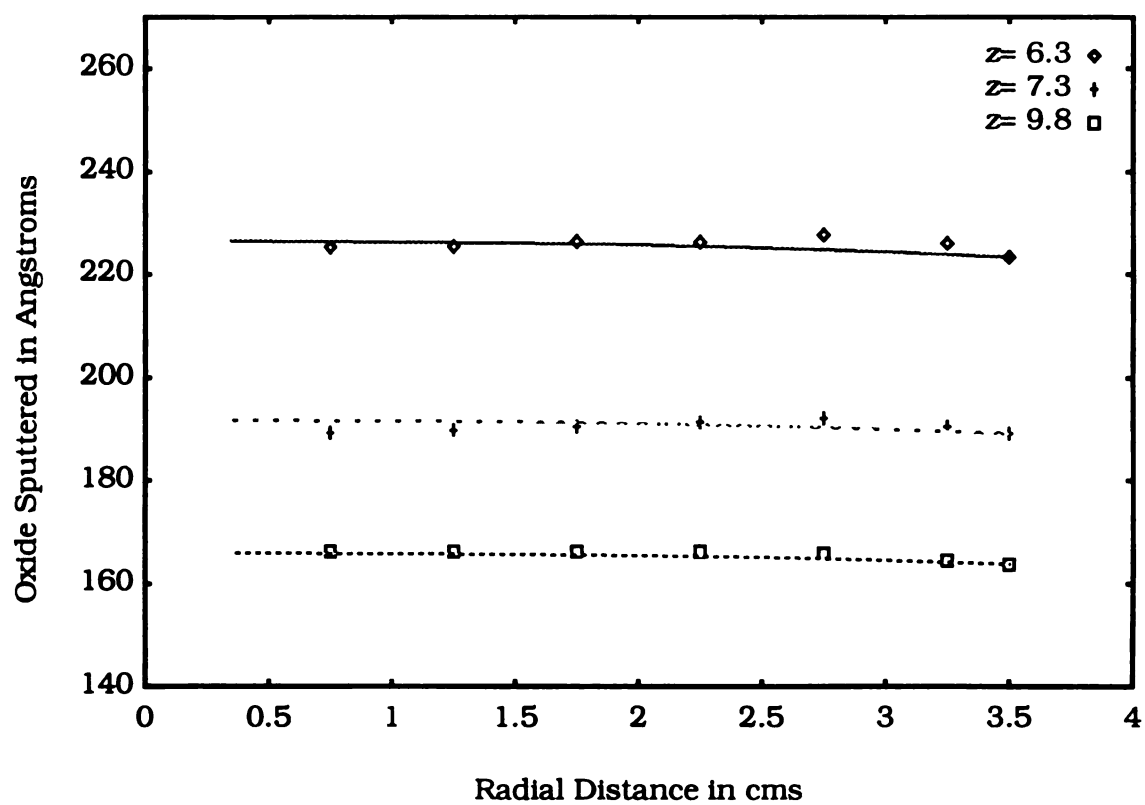


Figure 3.6: Comparison of theory with experiment for 3 downstream points.

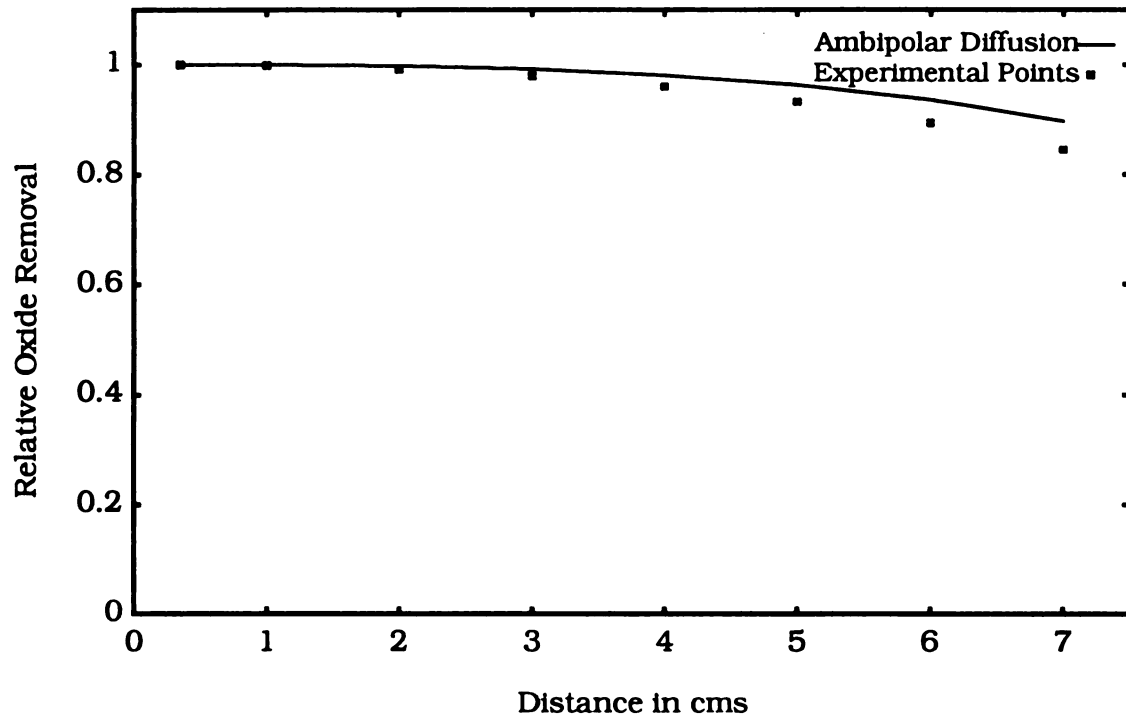


Figure 3.7: Comparison of theory with experiment for a 6 inch wafer.

3.1.4 Summary of Downstream Measurements

The sputter profile approximately follows the measured plasma density and the ambipolar diffusion calculated profiles. For a 9 cm diameter ECR discharge, the standard deviation percentage uniformity for 3 inch wafers was approximately 9%. For the 25 cm discharge system the uniformity across 3 inch wafers was 2-4% and less than 6% for 6 inch wafers. At the nominal conditions of 500 W microwave power, 9.8 cm downstream, -50 V RF induced bias, 1 mTorr pressure and 20 sccm argon flow the sputtering rate was about 40 Angstroms/min. The sputter rate increased, as expected, with increasing DC bias and increasing microwave power. The sputter rate decreases with increasing pressure, which may seem counter intuitive since plasma density increases with pressure. However, the rate decrease is attributed to the fact that plasma potential decreases with pressure thereby reducing the ion energies and sputter rate. The experimental sputter rates agreed approximately with the predictions of the ambipolar diffusion model in the radial and the downstream directions for both 3 inch and 6 inch wafers. Some discrepancy between the experimental data and the ambipolar diffusion model towards the edge of the wafers exists.

In conclusion, the sputter removal of oxide from silicon wafers was demonstrated with useful sputter rates and good uniformity using a 25 cm diameter ECR argon discharge. The ambipolar diffusion model of the plasma processing chamber proved useful for predicting the scaling properties of plasma reactors. However, the ambipolar diffusion model is limited in that it does not predict the plasma potential (i.e., ion energy) and it assumes a uniform plasma density existing at the source which is known to have non-uniformities.

It should also be noted here that the ambipolar model applied in this chapter is useful in estimating the discharge characteristics of downstream regions in the case of radial symmetry and the absence of confining magnets. In studies involving source regions and/or in the presence of magnets, a more complicated and detailed simulation detailed in subsequent chapters will have to be done.

3.2 Compact Ion Source

The compact ECR source shown in Figure 3.8 has been extensively characterized in [24]. This microwave cavity has an outer wall diameter of 5.8 cm and a height of 12.4 cm feeding an ECR plasma generation region of 3.6 cm internal diameter and approximately 3 cm height. The ECR zones are created by 3 annular ring magnets of pole strength 0.25 Tesla. Experimental measurements of a helium discharge [74] showed charge densities ranging between 3.8×10^9 - $6.2 \times 10^9/\text{cm}^3$ for pressures between 0.07 - 0.2 mTorr, input power between 90 - 115 Watts, and electron temperatures between 8 - 10 eV.

Field pattern measurements of the absolute magnitude and spatial variation of the impressed electric field have been conducted on this source using an array of holes drilled in its wall [75]. Measurements verify that a standing wave is created in the coaxial section of the cavity. The field is measured using a calibrated micro-coaxial probe which reads power from which the electric field is calculated. It has been shown that a maximum electric field of 40 kV/m can be obtained at 130 W input power. Investigations of this plasma source and another small diameter microwave plasma source by Mahoney [23], [76], have shown that these sources operate (in the pressure range (0.1 - 10 mTorr)) with a free fall diffusion behavior at lower pressures and with a Schottky diffusion behavior at

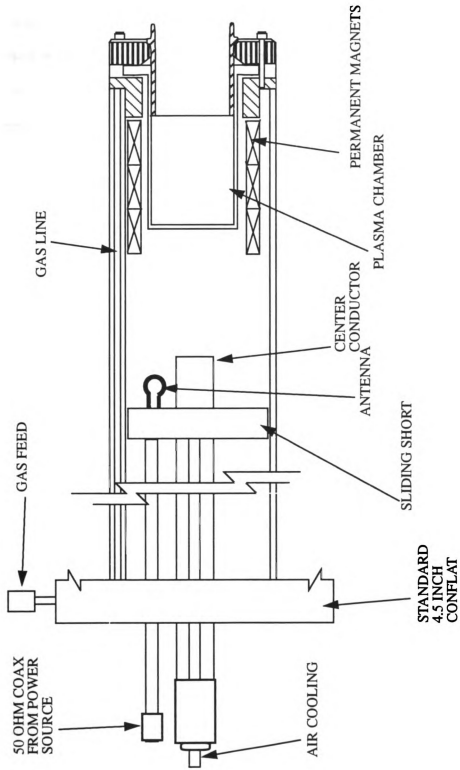


Figure 3.8: Compact ECR ion source.

higher pressures. The understanding and optimized design of these small microwave plasma sources has in the past been accomplished using knowledge and scaling techniques derived from the free-fall diffusion and Schottky diffusion models. As these sources are further studied and optimized, the limits of the application of these analytical models are reached and more comprehensive numerical models are needed.

Chapter 4

A 3D3V Plasma Simulation System

4.1 Introduction

This chapter introduces the model developed in this research to simulate plasma generation sources in three dimensions. The implementation of non-terminated grids to improve numerical stability is presented. The methodology used in developing a complete self-consistent EM-plasma model and the details of combining the two is also presented. Methods to take advantage of the inherent parallel nature in the simulations are also presented.

4.2 Design of the Plasma Simulation

Particle type plasma simulation are by nature very compute and memory intensive. The fundamental constraint of numerical simulation of plasmas is that the grid spacing in each of the simulated dimensions should be less than the Debye length λ_d (in cm) given by

$$\lambda_d = 743 \sqrt{\frac{T_e}{N_e}} \quad (4.1)$$

where T_e is the electron temperature in electron volts and N_e is the background charge density in cm^{-3} . This restriction presents fundamental problems on the size of plasmas that can be simulated on conventional desktop computers. For

example, a plasma at an electron temperature of 7 eV and charge density of $7 \times 10^9 \text{ cm}^{-3}$ corresponds to a Debye length of approximately $2 \times 10^{-2} \text{ cm}$. In order to simulate a small cavity of 1.8 cm radius and 3 cm length, one would require nearly 80 grids in the radial and 140 grids in the axial direction. With these restrictions, one cannot simulate the complete cylindrical structure on most conventional computers. Therefore, a simulation has to take advantage of the inherent symmetry of the structure as discussed in Section 2.4.

The iteration timestep of a plasma simulation is required to be $< 2\pi/\omega_{pe}$, where ω_{pe} is the plasma frequency in rad/sec given by

$$\omega_{pe} = \left(\frac{n_e q^2}{m_e \epsilon_0} \right)^{\frac{1}{2}} = 56.5 n_e^{\frac{1}{2}}, \quad (4.2)$$

q is the electron charge and m_e is the mass of the electron. Further, the timestep should be small enough that no particle should be able to traverse a distance larger than the Debye length. This ensures that Debye shielding phenomenon is maintained in the plasma being simulated.

4.2.1 A Non-Terminated Grid Design

The example given in the Section 4.2 leads to another inherent problem present in simulation in cylindrical co-ordinates. A large number of grids in the radial direction directly leads to a large difference in the volumes enclosed by the grids in the center to the ones further out in the radial direction. This can lead to large instabilities in the solution of plasmas in general and Poisson's equation in particular. In order to reduce such instabilities, a different grid scheme has to be

developed. This section introduces one such grid scheme developed for this research.

The main aim behind the development of this grid scheme is to ensure that the volumes enclosed by the grid points are of the same order of magnitude. This is accomplished by having different number of grid points in the ϕ directions at different radial points as can be seen in Figure 4.1. It can be seen that as the distance increases in the radial direction, the number of ϕ grids also increase leading to the equal volume design requirement. This sort of grid design presents new challenges in determining the grid point closest to the present location of a particle and in locating the closest grid points while performing CIC type calculations. Further, the elementary volumes for these grids also have to be calculated carefully so as to prevent incorrect charge density calculations.

In order to achieve this end, a linked list type neighbor scheme was designed in which each point maintains a list of all the indices of its neighbors in the three directions. In addition to these, the grid points in the radial region where there is a discontinuity in the ϕ direction also maintain the indices of the two closest grid points in the lower radial direction. The grid point that is closest to the particle in all three dimensions is considered the operating point for calculating the contribution of the particle to the grid and vice-versa. This calculation is the same as the conventional CIC calculation when the particle is in a cell between radial points with the same number of ϕ grids. However, when the particle straddles non-terminated grid boundaries as seen in Figure 4.2, extra computation has to be performed to calculate CIC contributions. In this case, the elementary cell contains 10 grid points instead of the conventional 8 and the location of the particle in the ϕ direction determines which half of the

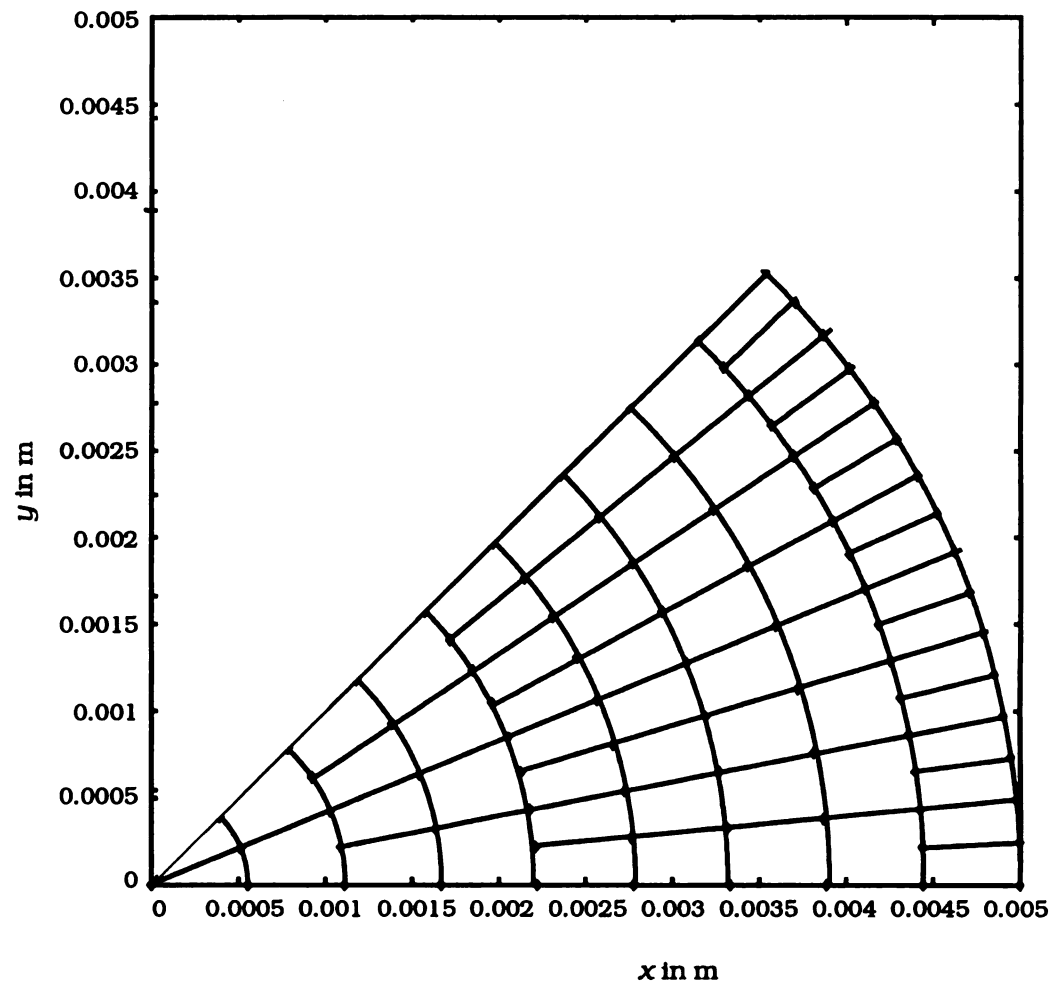


Figure 4.1: Non-terminated grid scheme.

elementary volume the particle is considered to be located as can be seen in Figure 4.2. In the figure, the location of the particle determines that the particle belongs to the volume created by the grid points 2, 4, 5, 7, 9,10 and the pseudo points x' and y' . The value of the potentials at the two pseudo grids points are interpolated from their respective ϕ neighbors.

This grid design proved to be very successful in solving serious problems of stability of the simulation at the center. In addition to this, this type of grid structure reduced the overall number of grids required to simulate the structure by over 50%. This allowed for the design of larger structures, or alternatively, higher density plasmas (smaller Debye length). However, it should be noted here that these advantages came at some computational overhead required in locating and maintaining the grid points.

4.2.2 Poisson's Equation Solution

Two dimensional $r\phi$ and rz finite-difference equations developed in [7] for Poisson's equation (2.5) can be extended into three dimensional cylindrical coordinates as follows

$$\begin{aligned} \frac{-\rho_{j,k,l}}{\epsilon} = & \frac{2r_{j+\frac{1}{2}}}{(\Delta r^2)_j \Delta r_{j+\frac{1}{2}}} (V_{j+1,k,l} - V_{j,k,l}) - \\ & \frac{2r_{j-\frac{1}{2}}}{(\Delta r^2)_j \Delta r_{j-\frac{1}{2}}} (V_{j,k,l} - V_{j-1,k,l}) + \frac{(V_{j,k,l+1} - 2V_{j,k,l} + V_{j,k,l-1})}{(\Delta z^2)_l} + \\ & \frac{2\Delta r_j}{(\Delta r^2)_j (\Delta \phi)^2 k r_j} (V_{j,k+1,l} - 2V_{j,k,l} + V_{j,k-1,l}) \end{aligned} \quad (4.3)$$

where $V_{j,k,l}$ is the potential at grid point (j,k,l) corresponding to the three directions (r,ϕ,z) , $(\Delta\phi)_k$ is the angle between grids $(j,k+1,l)$ and (j,k,l) , r_j is the radial distance to the j^{th} grid point, $r_{j+1/2} = 1/2 (r_{j+1} + r_j)$,

$$\Delta r_{j+\frac{1}{2}} = r_{j+1} - r_j \quad (4.4)$$

and

$$(\Delta r^2)_j = r_{j+\frac{1}{2}}^2 - r_{j-\frac{1}{2}}^2 \quad (4.5)$$

with the condition $j > 1$. Numerical solution of Poisson's equation in cylindrical coordinates gives rise to a peculiar problem at $j = 1$ corresponding to $r = 0$. In this case the k points in the ϕ direction are essentially the same point. Again Birdsall's [7] 2-D version can be extended, with $(\Delta r^2)_0 = r_{1/2}^2$, as

$$\begin{aligned} \frac{-\rho_{0,k,l}}{\epsilon} &= \frac{2r_{\frac{1}{2}}}{(\Delta r^2)_0 \Delta r_{\frac{1}{2}}} (V_{1,k,l} - V_{0,k,l}) + \frac{(V_{0,k,l+1} - 2V_{0,k,l} + V_{0,k,l-1})}{(\Delta z^2)_l} \\ Q_0 &= \sum_k \frac{V_{1,k,l} - V_{0,k,l}}{\Delta r_{\frac{1}{2}}} r_{\frac{1}{2}} \Delta\phi \end{aligned} \quad (4.6)$$

with the second part of the equation being a summation of all charge at origin. This equation is equivalent to the statement that all ϕ points at the origin are the same. E_r and E_z , the electric fields in the r and z directions, respectively are calculated similar to equation (2.32). However E_ϕ is calculated as

$$(E_\phi)_{j,k+\frac{1}{2},l} = -\frac{(V_{j,k+1,l} - V_{j,k,l})}{r_j (\Delta\phi)_k} \quad (4.7)$$

since $r_j \Delta\phi$ is the grid spacing in the ϕ direction. These equations are modified slightly for non-terminated grid points by assuming a pseudo point in the lower radial direction as described earlier.

4.2.3 Plasma Kinetic Model

A realistic simulation of plasma dynamics should include the collisions undergone by the different particles in the system. These collisions which can be broadly classified into elastic and inelastic types, lead to a change in the final state of the particle being considered. The different collision types that are considered in this simulation are:

1. Electron-neutral ionization collisions,
2. Electron-neutral momentum transfer collisions,
3. Electron-electron collisions,
4. Ion-neutral momentum transfer collisions, and
5. Ion-ion collisions.

Of the above listed collisions, all are considered elastic except for (1) which is inelastic leading to formation of new particles. Monte-Carlo methods introduced in Section 2.3.1.2 are used extensively to determine the occurrence of a collision and the final state of the particles after collision. The details of implementation of the collisions are explained in the sections that follow.

4.2.3.1 Particle-Neutral Collisions

One of the many collisions that can occur between a charged particle and neutrals is an elastic momentum transfer collision. This is a momentum randomizing event leading to the overall change of the particle velocity from a directed to a less directed state. This type of collision is very important in higher pressure plasmas wherein they play a major role in sustaining the plasma by collisional or joule heating. The momentum randomizing nature of these

collisions lead to a change in phase between the particle motion and the driving microwave field thus enabling the particle to gain energy from the field.

The general momentum transfer collision frequency is given by the formula

$$\nu_{me} = n_n \langle \sigma_m v \rangle \quad (4.8)$$

where σ_m is the cross section for momentum transfer and v is the velocity and $\langle \sigma_m v \rangle$ denotes the average value of the product over all energies. Plots of cross section vs. energy for ion-neutral and electron-neutral collisions for argon and helium can be found in [77]. Equation (4.8) is multiplied by the iteration timestep, Δt , to obtain the number of each type of collisions per iteration. These numbers are used in the Monte-Carlo method to determine the occurrence and type of collision as described earlier.

The determination of post-collision velocity components of the particle also makes extensive use of random numbers. Since this is an elastic collision, the total energy of the particle does not change, however, the particle loses some of its directionality. Let v_x , v_y and v_z be the velocity components of the particle before any collision and let r_1 and r_2 be random numbers between 0 and 1. The scattering angles in the azimuth and the elevation are then calculated as follows

$$\begin{aligned} \phi &= 2\pi \times r_1 \\ \theta &= \arccos(1 - 2r_2) \end{aligned} \quad (4.9)$$

and then the new components of the velocity, v_x' , v_y' and v_z' , are calculated as follows

$$\begin{aligned} v_x' &= V_{mag} \times \cos \phi \times \sin \theta \\ v_y' &= V_{mag} \times \sin \phi \times \sin \theta \\ v_z' &= V_{mag} \times \cos \theta \end{aligned} \quad (4.10)$$

where $v_{mag} = \sqrt{v_x^2 + v_y^2 + v_z^2}$ is the absolute value of the velocity before and after the collision.

4.2.3.2 Charged Particle-Particle Collisions

Charged particle collisions of the type (3) and (5) are governed by long range coulombic interactions and cross sections for momentum transfer for collisions of this form are given in [6]. The calculation of the final states from the initial states of the two particles is done similar to the method detailed by Jacoboni and Luighi [32]. The calculation of the final states \vec{k}' and \vec{k}'_0 from the initial states \vec{k} and \vec{k}_0 are done by first calculating the initial and final relative wave vectors, \vec{g} and \vec{g}' , respectively as follows

$$\begin{aligned}\vec{g} &= \vec{k} - \vec{k}_0 \\ \vec{g}' &= \vec{k}' - \vec{k}'_0\end{aligned}\tag{4.11}$$

The azimuthal angle between \vec{g}' and \vec{g} is taken at random between 0 and 2π , the elevation angle is calculated as

$$\cos\theta = \frac{2r}{1 + g^2(1-r)/\lambda_d^2}\tag{4.12}$$

where r is a random number between 0 and 1, $g = |\vec{g}| = |\vec{g}'|$ and λ_d is the Debye length. Finally the new states, \vec{k}' and \vec{k}'_0 , of the particles are calculated as

$$\begin{aligned}\vec{k}'_0 &= \vec{k}_0 - 0.5(\vec{g}' - \vec{g}) \\ \vec{k}' &= \vec{k} + 0.5(\vec{g}' - \vec{g})\end{aligned}\tag{4.13}$$

The collision cross-sections and frequencies for these type of collisions are tabulated in [78],[79].

4.2.3.3 Electron-Neutral Ionizing Collisions

Unlike the collisions discussed earlier, ionizing collisions are inelastic. After an ionizing collision, the original electron involved in the collision loses the amount of energy required for ionization, termed as ionization potential E_{th} , and results in the creation of a new electron-ion pair. This type of collision is the source of fresh charged particles to compensate for those lost due to different mechanisms. The values of the ionization rates depends on the background neutral density and uniquely defines the qualities of the plasma with respect to the power required to maintain a certain density. A typical ionization cross section curve is shown in Figure 4.3. As can be seen in the figure, the magnitude of the cross section changes with the energy of the electron showing an initial increase and then a decrease with respect to energy. The magnitude of E_{th} for argon and helium are 15.76 eV and 24.588 eV, respectively. The collision frequency is calculated by numerically integrating the equation

$$\nu_i = \frac{8\pi}{m_e^2} \int_{E_i}^{\infty} \sigma(E) F(E) E dE \quad (4.14)$$

where E is the energy, $\sigma(E)$ is the collision cross section as a function of energy and $F(E)$ is the distribution function. In particle type simulations, the number of ionizing collisions occurring per iteration is calculated as follows

$$\begin{aligned} \nu_i &= n_n \sigma(E) V_{mag} \\ N_{iter} &= \nu_i \Delta t \end{aligned} \quad (4.15)$$

where n_n is the neutral density, V_{mag} is the magnitude of the velocity of an electron with energy greater than E_{th} , Δt is the iteration timestep and N_{iter} is the number of ionizing collisions per iteration. If this number is less than 1, a

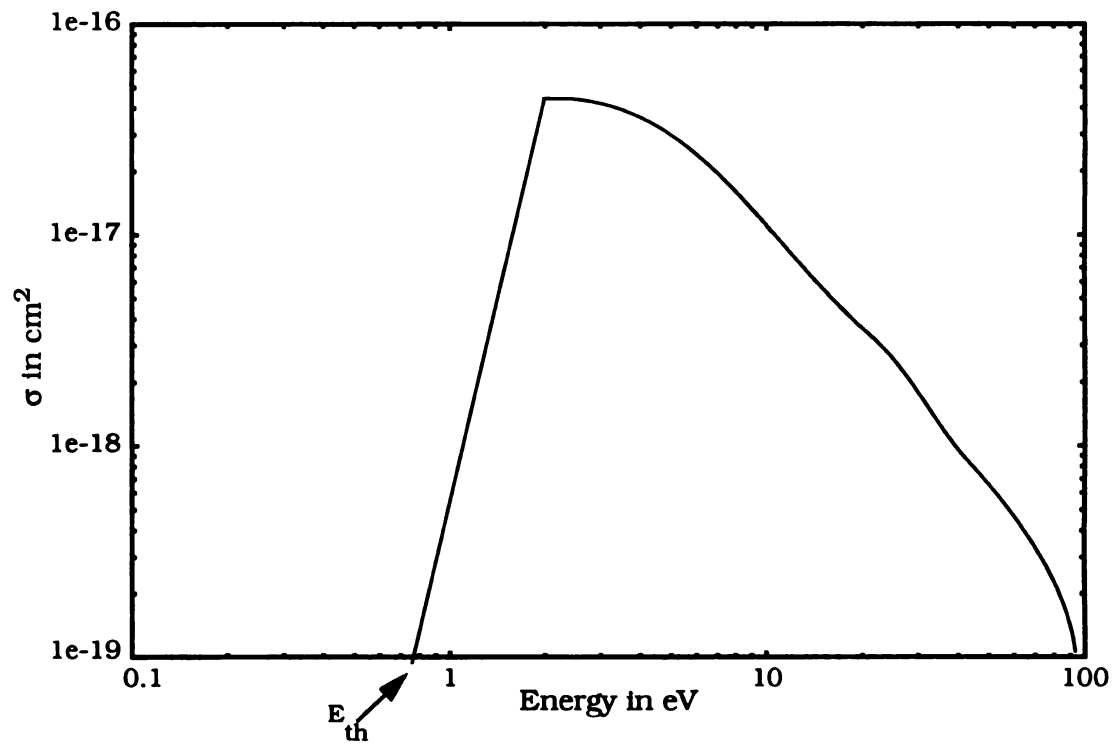


Figure 4.3: Example ionization cross section plot.

random number is generated and an ionization is assumed to occur if its value is less than N_{iter} . If the value N_{iter} is greater than 1, an integral number of new electron-ion pairs are created and the remaining fraction is treated using the Monte Carlo method.

Ionizing collisions result in the formation of a new electron-ion pair and in the reduction of the total energy by the amount equal to the ionization potential of the background gas. The total energy of the three particles after the collision, E_{tot} , is given by

$$E_{tot} = E_{old} - E_{th} \quad (4.16)$$

where E_{old} is the pre-collision energy of the electron causing the ionization. Since the average energy of electrons is much higher than that of the ions and the neutral gas, it is assumed that the newly created ion of the electron-ion pair is essentially at rest, and E_{tot} is distributed randomly between the two electrons as follows

$$\begin{aligned} E_1 &= E_{tot} \times r_1 \\ E_2 &= E_{tot} (1 - r_1) \end{aligned} \quad (4.17)$$

where E_1 and E_2 are the post-collision energies of the two electrons and r_1 is a random number between (0,1). The velocities of the two electrons in each of the three directions is calculated similar to equations (4.9) and (4.10).

4.2.4 Maxwell's Equation Solution

A complete 3D3V simulation of microwave plasmas involves the self-consistent coupling of Maxwell's equation with particle motion. The microwave \mathbf{E} field driving the plasma is calculated using a Finite Difference Time Domain

technique (FDTD) to calculate the fields at the next iteration timestep. This involves solving equations (2.1) and (2.2) on the simulation grid structure. The plasma acts as a load providing the current in equation (2.2). The design and details of the Maxwell's equation solver is discussed in considerable detail in [41].

4.2.5 Self-Consistent 3D3V Plasma Simulations

A self-consistent 3D3V plasma simulation involves solving plasma dynamics along with Maxwell's equation. While individually the plasma and Maxwell equations simulation can be done with reasonable effort, the combined simulation provides new problems. In some cases, the solution of Maxwell's equation and Poisson's equation have contradicting requirements and as a result the combination of the two can prove challenging. The following sections detail the methods to successfully accomplish a self-consistent plasma simulation.

4.2.5.1 Geometrical Requirements

The fundamental requirement of a Maxwell's equation solver is that the values of the grid spacing and the timestep be such that the EM wave being solved does not advance more than a grid distance per iteration. Mathematically, this requirement for a solution in rectangular co-ordinates is given by

$$\Delta t < \frac{1}{c \left(\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} \right)}. \quad (4.18)$$

where c is the speed of the EM wave, Δt is the iteration timestep and Δx , Δy and Δz are the grid spacing in the x , y and z directions, respectively. A further restriction on Δt is that it should be less than the value of the EM wave's time period.

These requirements dictate that a stable Maxwell's equations solver have a very small timestep and a relatively larger grid spacing as compared to the plasma simulation requirement. Simulations wherein Maxwell's equation timestep is a thousandth of the plasma timestep and its grid spacing is four times the plasma grid spacing are quite common. This mismatch becomes worse when simulating a higher density (smaller Debye length) plasma.

Further, the Maxwell's equation solver designed by [41] solved the equations in regular cylindrical coordinates whereas the plasma simulation used the staggered grids introduced in Section 4.2.1. The use of different grid structures leads to the requirement of interpolation and extrapolation of the data between the two programs.

In addition to all these requirements, there is a hidden instability in the discrete solution of Maxwell's equations. For example, the solution for E_r , the radial component of the electric field, at the grid point $(i+1/2, j, k)$ is given by

$$E_r^{n+1}\left(i+\frac{1}{2}, j, k\right) = E_r^n\left(i+\frac{1}{2}, j, k\right) + \frac{\Delta t}{\epsilon\left(i+\frac{1}{2}, j, k\right)} \quad (4.19)$$

$$\times \left(\frac{H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right) - H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)}{r_{i+\frac{1}{2}} \Delta \phi} - \frac{H_\phi^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) - H_\phi^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right)}{\Delta z} - J_r^n\left(i+\frac{1}{2}, j, k\right) \right)$$

where H_z and H_ϕ are the z and ϕ components of the magnetic field and J_r is the radial component of the current. In the above equation the current term can be modified by using an equivalent conductivity of the form

$$J_r^n\left(i + \frac{1}{2}, j, k\right) = \sigma_r^n\left(i + \frac{1}{2}, j, k\right) \times E_r^{n+1}\left(i + \frac{1}{2}, j, k\right). \quad (4.20)$$

to give a modified form of equation (4.19) as follows

$$E_r^{n+1}\left(i + \frac{1}{2}, j, k\right) = E_r^n\left(i + \frac{1}{2}, j, k\right) \times \left(1 - \sigma_r^n\left(i + \frac{1}{2}, j, k\right) \times \frac{\Delta t}{\varepsilon\left(i + \frac{1}{2}, j, k\right)}\right) \cdot \quad (4.21)$$

$$\times \frac{\Delta t}{\varepsilon\left(i + \frac{1}{2}, j, k\right)} \left(\frac{H_z^{n+\frac{1}{2}}\left(i + \frac{1}{2}, j + \frac{1}{2}, k\right) - H_z^{n+\frac{1}{2}}\left(i + \frac{1}{2}, j - \frac{1}{2}, k\right)}{r_{i+\frac{1}{2}} \Delta \phi} - \frac{H_\phi^{n+\frac{1}{2}}\left(i + \frac{1}{2}, j, k + \frac{1}{2}\right) - H_\phi^{n+\frac{1}{2}}\left(i + \frac{1}{2}, j, k - \frac{1}{2}\right)}{\Delta z} \right)$$

Since the instantaneous current in the plasma region is calculated using equation (2.35), a momentary noise in the current calculated using $\sigma_r^n\left(i + \frac{1}{2}, j, k\right) \times (\Delta t) / \left(\varepsilon\left(i + \frac{1}{2}, j, k\right)\right)$, leads to a value greater than unity which can produce instabilities. This requirement also dictates that the Maxwell's equation timestep be as small as possible to aid stability of the simulations.

4.2.5.2 Combined Plasma-Maxwell Simulation

In view of the conflicting requirements of the two parts of the simulation, it was decided to keep the two programs separate and to communicate only the relevant details between the two programs. The plasma program calculates the instantaneous currents at each grid point and interpolates them onto the

Ma

re

ph

de

at

ti

re

a

op

W

i

a

S

b

P

N

.

Maxwell's equation grid system and the Maxwell's equation solver calculates the new \mathbf{E} and \mathbf{H} fields using the current and communicates the new values to the plasma simulation as can be seen by the flow chart in Figure 4.4. The vertical dotted line divides the plasma and the EM subprograms. The horizontal dashed arrows signify the movement of data between the two subsystems. The different timestep requirements of the two programs has another effect on the overall time required for the simulation. As can be seen in Figure 4.4, the EM program does a number of iterations for every iteration of the plasma program. Therefore an optimum trade-off between speed of convergence and stability has to be located very carefully. Further, if there is a large difference in the magnitudes of the two timesteps, a small noise in current introduced to the EM program tends to get amplified over the iterations as explained in the previous chapter.

The grid currents are calculated using Marder's correction as explained in Section 2.3.4. PVM routines were used to synchronize and communicate data between the two programs. It should be noted here that the nature of the program is strictly serial, i.e., the two programs alternate in their execution and PVM is used here only as a data communication tool. The two programs use non-blocking receive to ensure strict serial fashion of execution flow and communicate using file I/O. The EM program writes to a mutually agreed file and sends a pre-arranged message to the plasma program which reads the file upon receipt of this message and vice-versa.

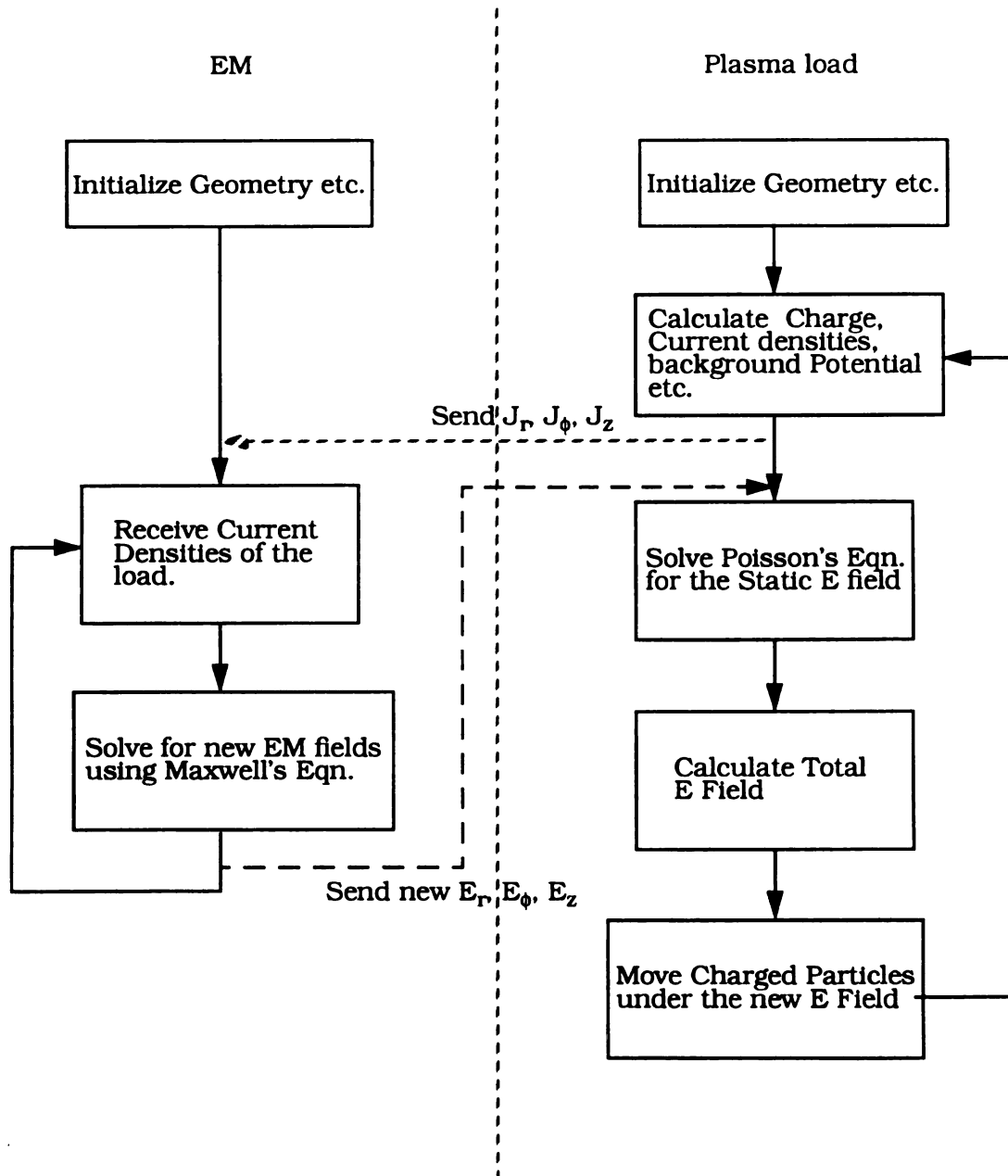


Figure 4.4: Combined plasma-EM simulation flowchart.

4.2.5.3 Power Absorption Calculations

Power absorbed by the plasma can be calculated in many different ways. One way is to calculate the value of the Poynting integral at a plane immediately above the quartz plane. The Poynting integral is written as

$$P_{in} = \oint (\mathbf{E} \times \mathbf{H}) \cdot d\mathbf{s}. \quad (4.22)$$

where P is the power in watts and $d\mathbf{s}$ is the area in the xy plane immediately above the quartz chamber.

The instantaneous power absorbed by the plasma can also be calculated by summing the product of current and field strength at every grid point. This is equivalent to the equation

$$P_{abs} = \int_v (\mathbf{J} \cdot \mathbf{E}) dv = \sum_{i,j,k} (J(i,j,k) \cdot E(i,j,k)) \times \Delta v(i,j,k) \quad (4.23)$$

where $\Delta v(i,j,k)$ is the incremental volume at the grid point (i,j,k) .

Another measure of the power required to sustain a plasma is the loss of energy from the electron gas due to ionizations and loss of particles to the walls.

This can be written in the form of an equation as

$$E_{loss} = \sum_{lost} energy + \sum_{ionizations} E_{th} \quad (4.24)$$

where E_{th} is the ionization threshold and E_{loss} is the energy lost in Joules. The power lost is calculated by averaging the energy loss over a microwave cycle. In

steady

power.

4.3 F

Th

natur

soluti

progr

In vie

simul

4.3

ch

co

c

R

pr

steady state the loss of power from the plasma is compensated by the input power.

4.3 Efforts to Speed up the Simulation

The combined 3D3V simulations tend to be very CPU intensive and the serial nature of the program shown in Figure 4.4 adds the time taken to converge to a solution. The smaller EM timesteps needed to ensure stability lead to the EM program spending considerably more time executing than the plasma program. In view of these problems, it is imperative that efforts be made to speed up the simulations. This section presents two such methods devised in this research.

4.3.1 Cycling the EM Program

It was theorized that since a stable plasma solution does not show rapid changes in behavior but instead shows a steady move towards a solution, considerable speed gains could be made by “turning” the EM program on and off. The EM program was let to run a certain number of integral microwave cycles and during this time the plasma program stored all components of the electric field for every plasma iteration. The EM program was then paused while the plasma program used the stored values of the fields to move particles around for the next few cycles. Then the EM program was restarted with the new values of the current. It was thought that the new values of the current calculated would allow the EM program to get back in synchronization with the plasma program.

0

opti

vers

that

sin

the

con

pr

pa

as

0

v

i

One of the major aspects of this implementation is the decision on the optimal number of cycles required to keep the results the same as the serial version and yet obtain the maximum speed-up possible. Initially it was thought that cycling the EM program every microwave cycle would be the ideal choice since this allows the EM program to be in relatively more communication with the plasma program. The accuracy of the implementation was decided by comparing the time evolution of particle and field characteristics of the cycled programs with their serial counterpart. In particular, the time evolution of particle number density, particle energy and EM field strength were considered as indicators of accuracy.

The time evolution of the absolute \mathbf{E} field strength ($\sqrt{E_r^2 + E_\phi^2 + E_z^2}$), averaged on all the grids at every iteration timestep, for the cases when the EM program was cycled on and off every cycle and every 3 cycles was identical as can be seen in Figure 4.5 for the plasma source described in Section 3.2. Note that the three plotted lines in the figure lie on top of each other and are indistinguishable. However, the time evolution of particle quantities present a different picture as can be seen from Figures 4.6 and 4.7. It can be clearly seen that the choice of cycling off and on every cycle is clearly the inaccurate one since it leads to considerable error in electron (most easily effected) number and energy. However, the choice of cycling the EM source every 3 cycles gives results that closely track the serial version. Clearly there is an optimal number between too many and too few cycles to ensure accuracy.

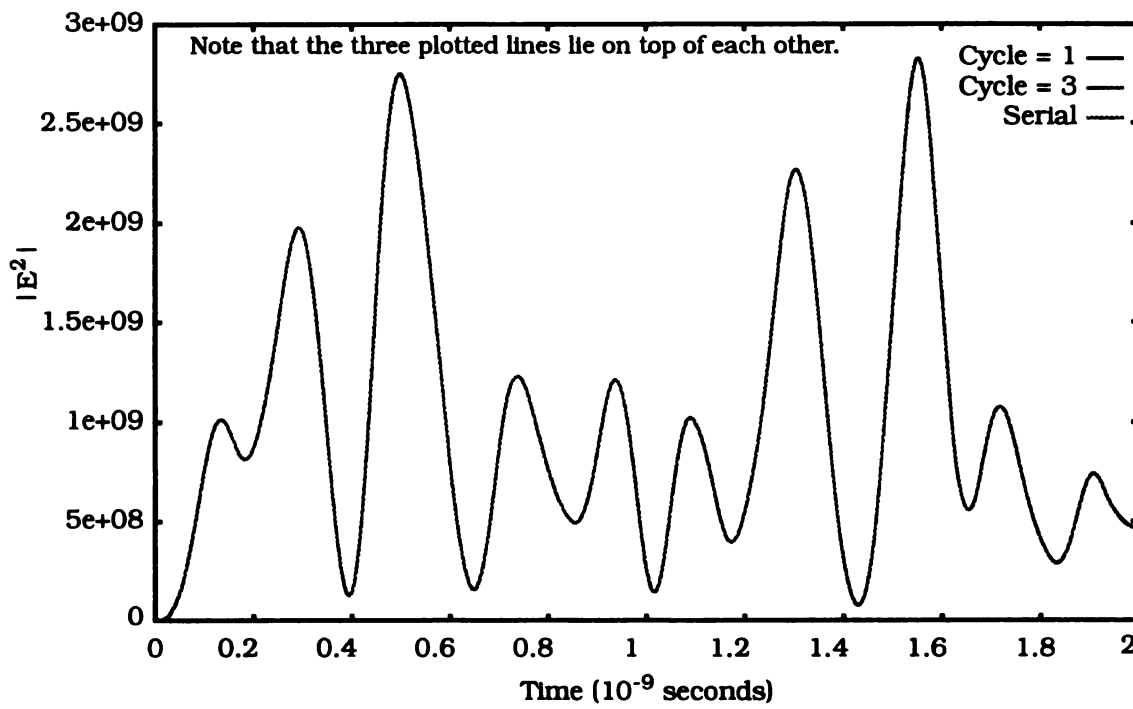


Figure 4.5: Time evolution of fields for the serial and cycled cases.

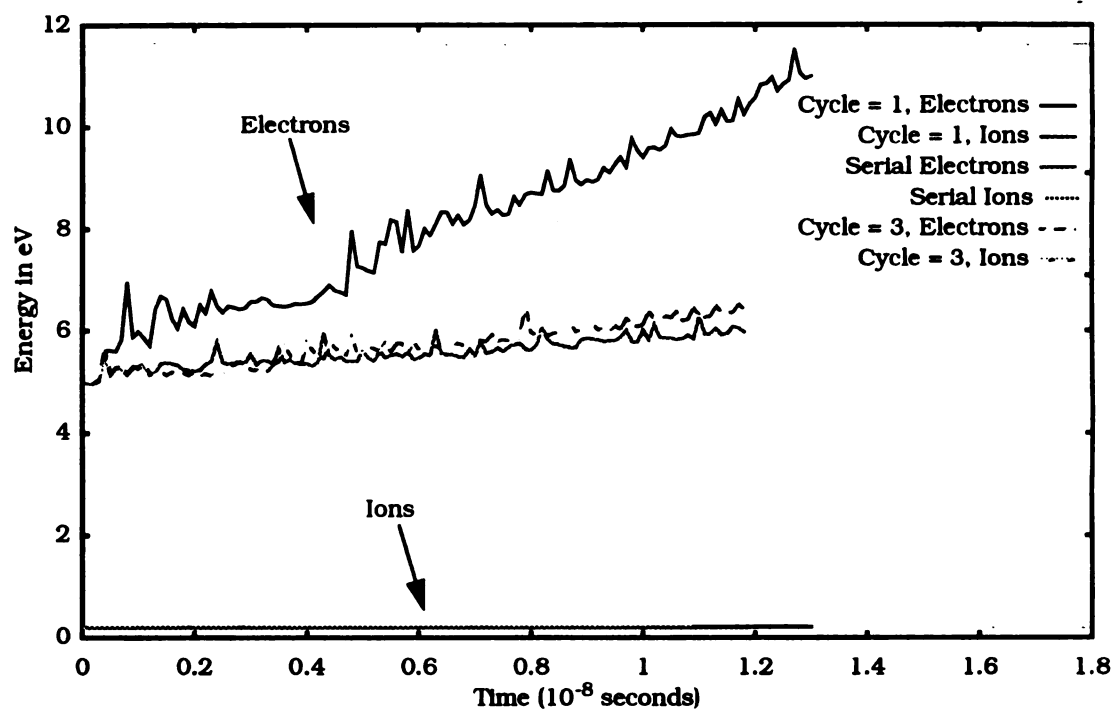


Figure 4.6: Energy of particles with source cycled on and off.

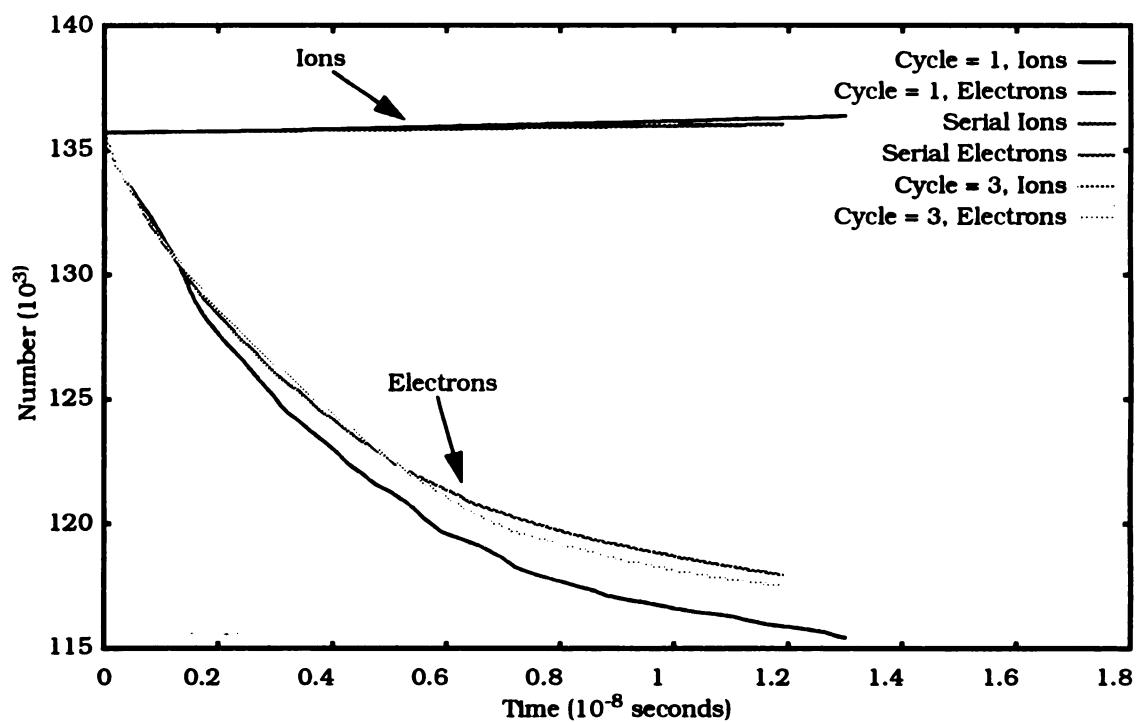


Figure 4.7: Number density of particles with source cycled on and off.

4.3.2 Implementation of a Parallel EM-Plasma Version

The flowchart shown in Figure 4.4 is strictly serial and it was theorized that some speed-up may be obtained by modifying the flowchart as shown in Figure 4.8. Instead of following the serial fashion of the complete simulation, this version theorizes that a phase difference of one timestep may correct itself over the complete run of the simulation. The flowchart in Figure 4.8 is identical to the one in Figure 4.4 except for the coupling between the two programs. Instead of following a strictly serial fashion where the EM program calculates the new fields, then sends the fields to the plasma program which moves the particles and calculates the new current components and so on, this version forces the EM code and the plasma code to execute in parallel. The two programs use initial values and continue their execution and at the end of each iteration, exchange data. This leads to the EM program generating the new fields from the load current values calculated for the fields generated in the previous step. Similarly, the plasma program uses the fields generated for the earlier timestep to move the particles. This leads to the plasma and EM program to be out of phase by one timestep. The electric fields generated by both these versions were identical as can be seen in Figure 4.9. Further, the time evolution of particle number density and particle energy closely track the serial case as can be seen in Figures 4.10 and 4.11, respectively.

It is interesting to note here that even though the two programs appear to execute in parallel, the speed gain is dictated by the slower program. For example, if the EM program iterates 1000 times for each plasma timestep, the time taken by the EM program dictates the overall speed. The plasma program

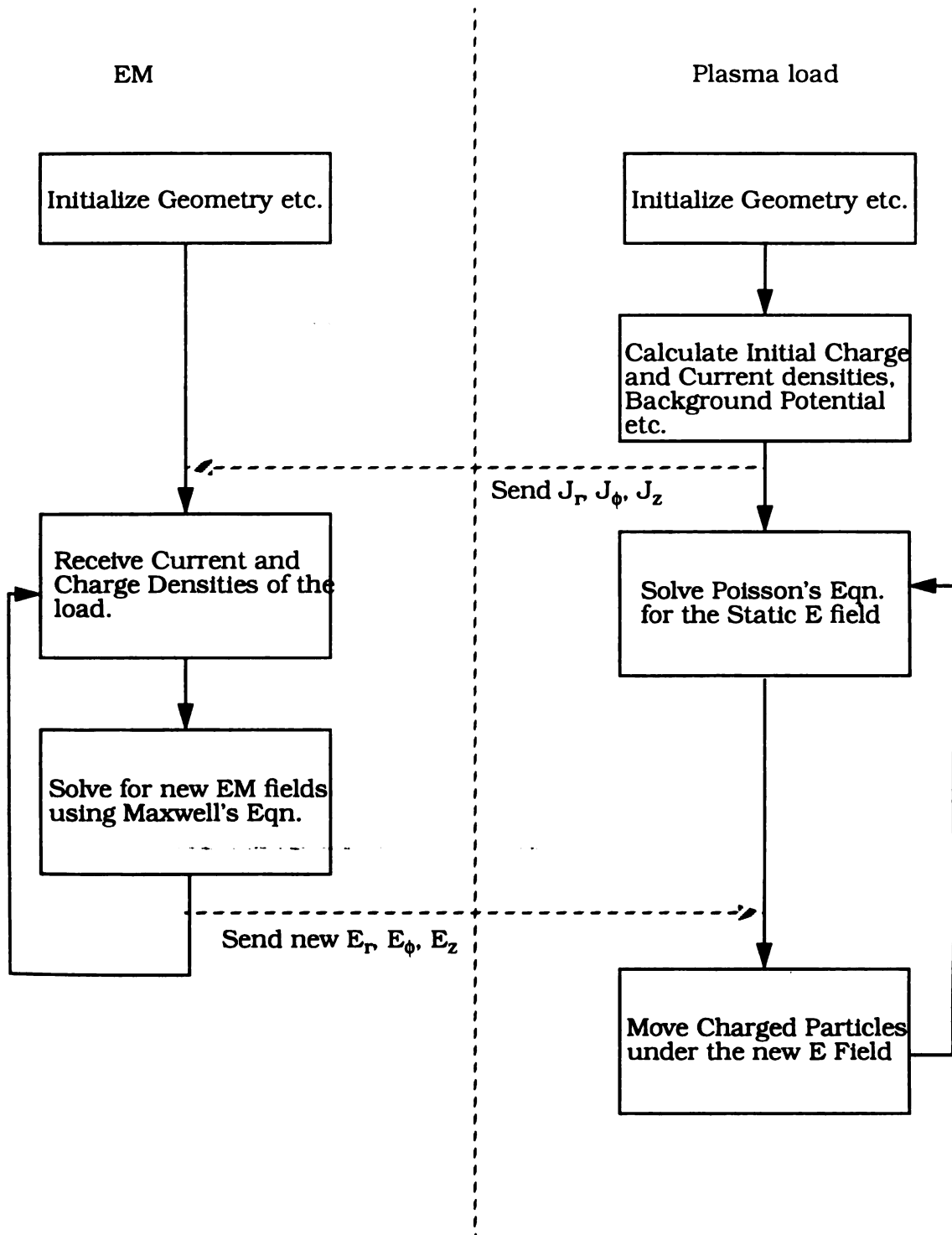


Figure 4.8: A parallel EM-plasma implementation.

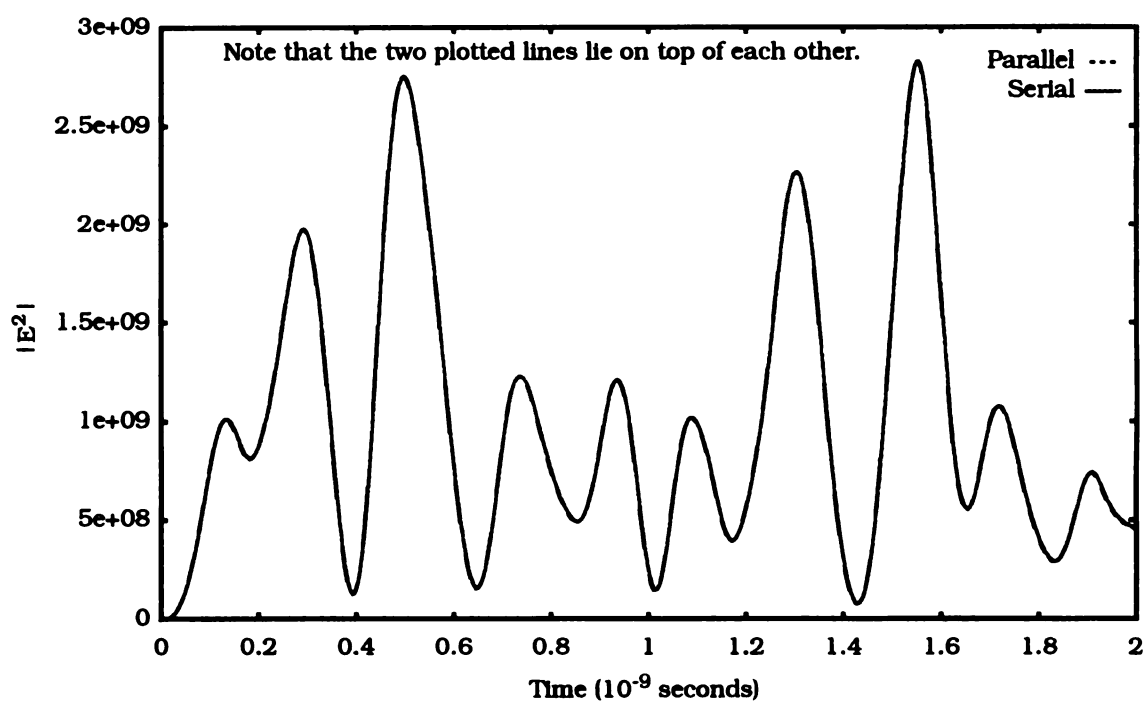


Figure 4.9: Time evolution of fields for the serial and parallel cases.

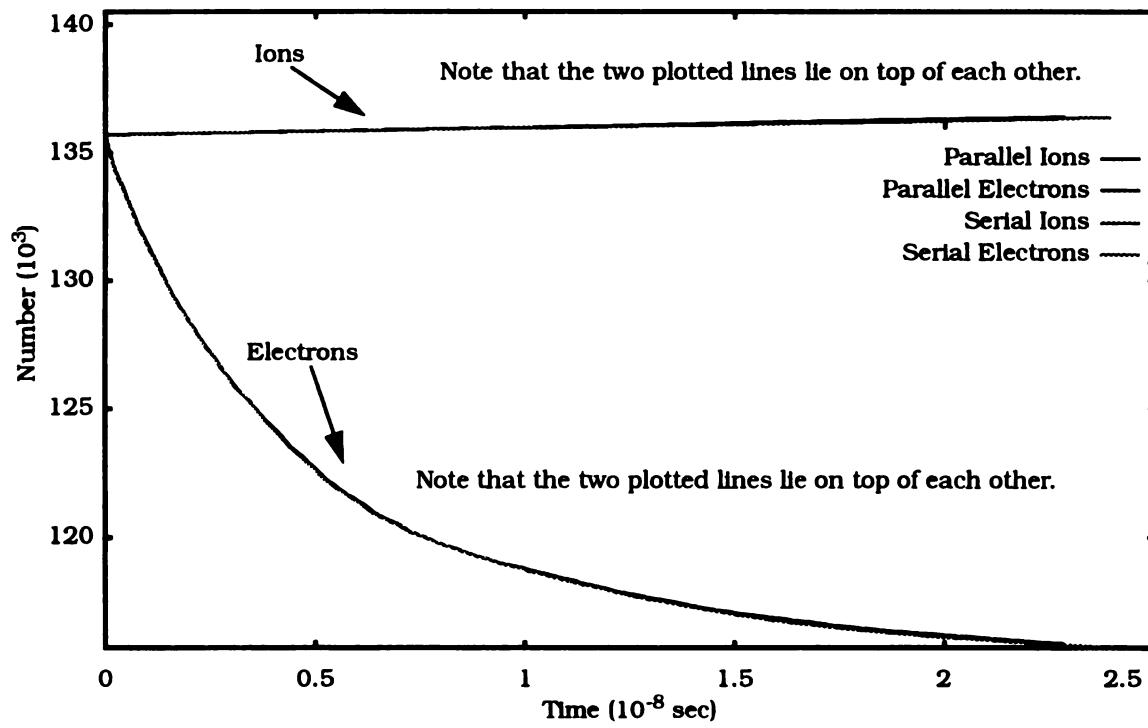


Figure 4.10: Evolution of number density for the serial and parallel cases.

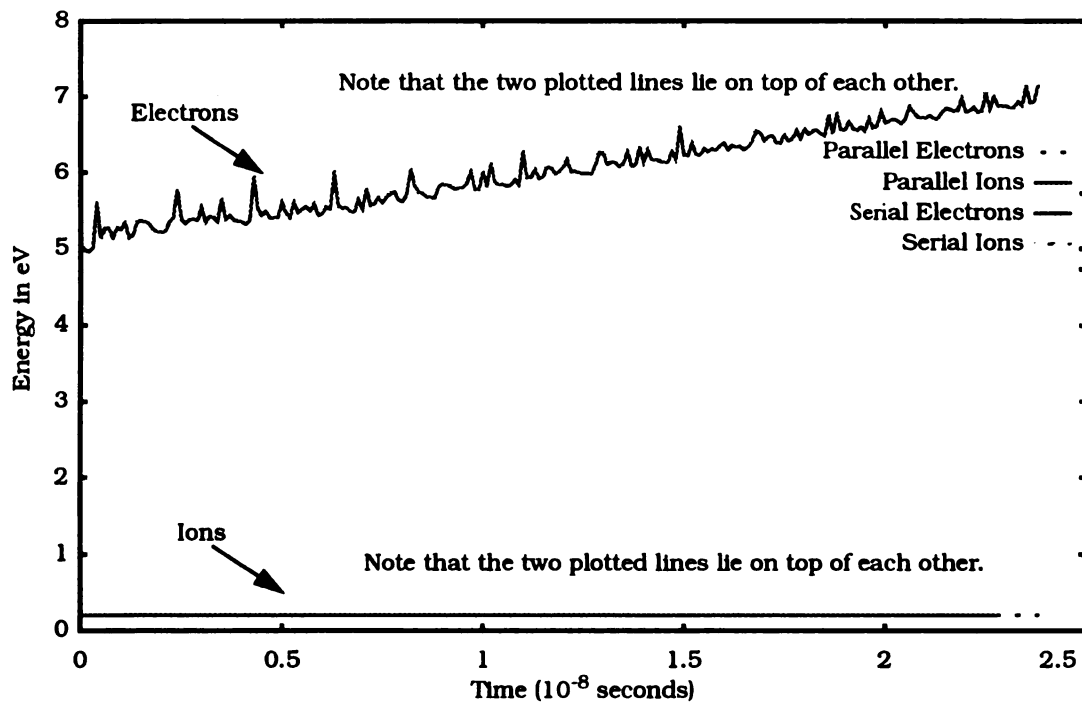


Figure 4.11: Time evolution of energy for the serial and parallel cases.

would have finished one iteration and would wait for data from the EM program. The two sub-systems have to be carefully designed to take full advantage of the parallel nature of this algorithm.

Chapter 5

Downstream Plasma Simulations

5.1 Introduction

A subset of the plasma-EM simulation model of the previous chapter is applied to the modeling of the downstream plasma processing region in this chapter. This chapter will demonstrate the techniques developed in Chapter 4 as they apply to the modeling problem described earlier in Chapters 2 and 3.

The downstream region of plasma processing chambers of the type shown in Figure 1.1 is essentially microwave field free and is controlled by space charge fields. The particles generated in the source region flow into the large processing region and expand to fill the volume. In the process some particles are lost to the walls leading to a build up of a plasma sheath. Any material placed in this region will interact electrically and chemically with the plasma leading to sputtering and deposition as the case may be.

This region when occupied by a low pressure plasma can be simulated using a particle type method. In the absence of non-uniform B_ϕ fields, this region can be simulated in two dimensions. This chapter introduces a 2D3V simulation of the downstream region of a typical plasma processing chamber. The use of different weighting factors for the charged particles in different regions to assist stability is also detailed.

These types of simulations can prove useful in predicting the final processing characteristics by providing an estimation of the spatial variation and magnitude of the different plasma variables like charged particle flux, densities and temperatures.

5.2 2D3V Downstream Simulation

The downstream region is considered uniform in the ϕ coordinate and the simulations are done in the rz coordinate system. The area being simulated can be seen in Figure 5.1. It is assumed that there is a fresh influx of charged particles from the source region that compensates for the loss of particles to the walls. The particles are added to the simulation at a rate calculated from the ion flux density given by the equation [80]

$$\Gamma_i = 0.61N_0 \left(\frac{k_b T_e}{M_i} \right)^{1/2} \quad (5.1)$$

where Γ_i is the average ion flux, N_0 is the constant ion density at the quartz opening to the downstream region, and M_i is the mass of the ion. Usage of equation (5.1) assumes that flux exiting the opening in the quartz tube reaches the walls in the downstream region and that the electron flux is equal to the ion flux. If the area of the quartz opening is A and the timestep is Δt then the average number of particles to be added per iteration is given by $\Gamma_i A \Delta t$. Since the number calculated in this fashion can be a fraction, the Monte-Carlo method is used to determine the number of particles added per iteration. Each “addition” indicates the creation of an electron-ion pair. These pairs are given random locations in space within the area of the source opening and a Maxwellian

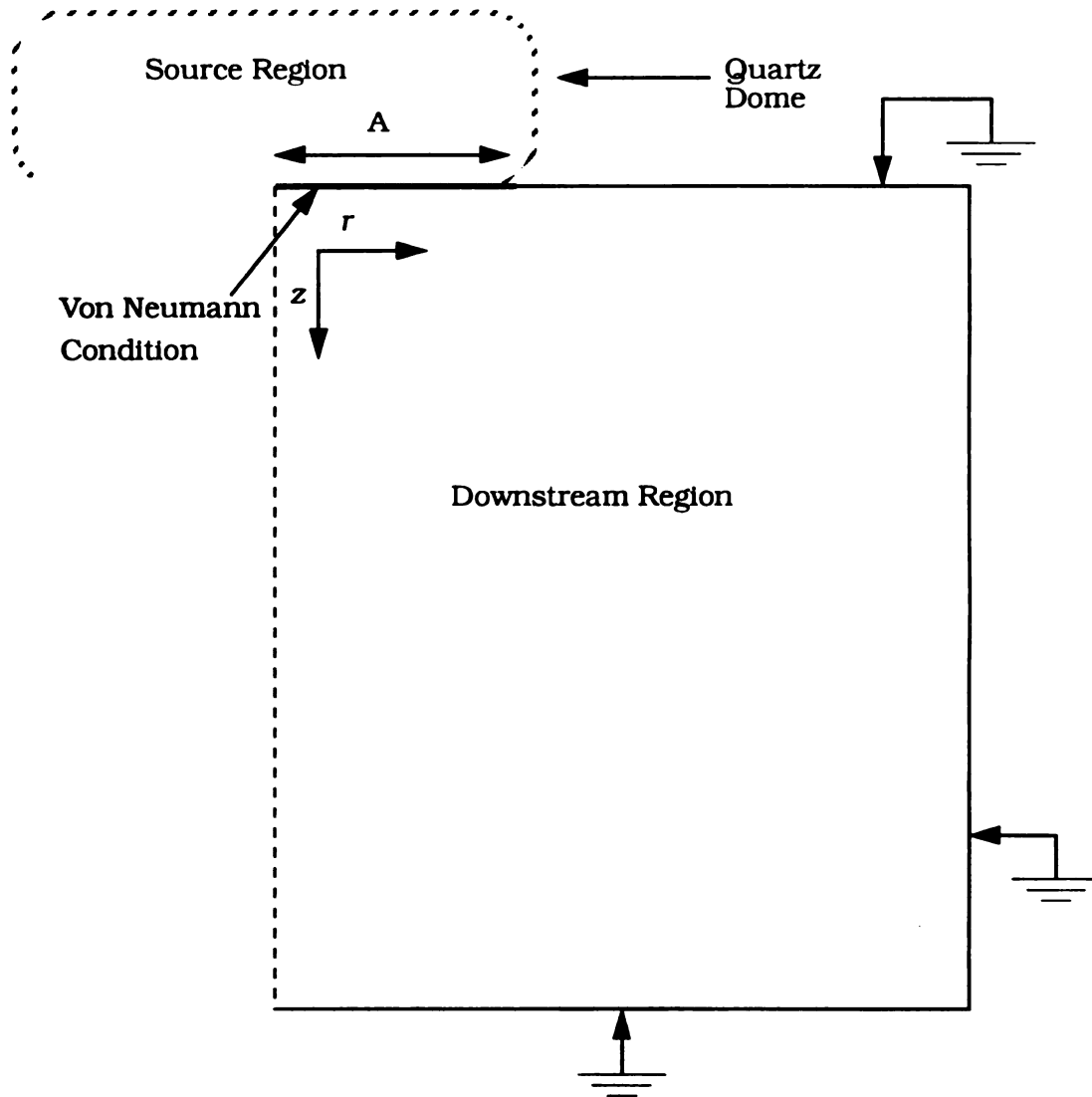


Figure 5.1: Downstream simulation zone.

distribution in energy is assigned based on the assumed distribution temperature. Further, the source boundary is considered reflecting such that any particle going into the source region is reflected back. The boundary conditions assumed can also be seen in the figure. The walls are assumed to be conducting and grounded. The Poisson equation boundary condition at the particle entrance boundary is a normal electric field of zero.

5.2.1 An Unequal Weighting Scheme

The non-terminated grid design for cylindrical structures introduced in Section 4.2.1 uses unequal number of grids in successive ϕ locations to aid Poisson's equation stability in the center. In the 2D case, the absence of the ϕ dimension makes it impossible to apply non-terminated grids. However, the problem of the successively larger areas for outer grid points remains. Further, an initially uniform distribution of particles gives considerably fewer particles in the grid points closer to the center. These two problems lead to instability while solving Poisson's equation.

Plasma simulations use a weighting scheme to determine the amount of charge carried by a *superparticle*. The simplest weighting scheme involves providing equal weights for all *superparticles* as given by

$$W = \frac{Vol \times N_e}{N_{tot}} \quad (5.2)$$

where W is the weight of the *superparticle*, N_e is the charge density, N_{tot} is the total number of *superparticles* and Vol is the total volume of the structure being simulated.

This uniform weighting scheme leads to instabilities at grid points closer to the center since there are fewer particles in this region and the movement of some particles out of the region leads to large fluctuations in charge density and hence in the solution of Poisson's equation.

In order to reduce these problems a non-uniform weighting scheme [81] has been implemented in this research. This scheme involves dividing the simulation area into a number of strips of areas that have similar size as can be seen in Figure 5.2. In cylindrical co-ordinates, this leads to strips closer to the center having more grid points. The weights of the particles are dictated by the region in which they lie. This scheme leads to particles being located in the strips closer to the center carrying correspondingly smaller weights. In this case the particle weights are calculated as

$$W(i) = \frac{Vol(i) \times N_e}{N_{tot}(i)} \quad (5.3)$$

where $W(i)$ is the weight of the *superparticle* in strip i , N_e is the charge density, $N_{tot}(i)$ is the total number of *superparticles* in the strip and $Vol(i)$ is the volume of the strip i . Particles are distributed uniformly in space in each of the strips. This equation reduces to the simple equation (5.2) when the weights are all equal.

This weighting scheme has the advantage of reducing the instabilities in the grids closer to the center. However, the computational overhead of the implementation increases considerably due to the discontinuous nature of the scheme. Particles crossing strip boundaries have to be compensated for correctly. Monte-Carlo methods are used extensively to determine the status of particles crossing strip boundaries. Particles crossing from a lower weighting

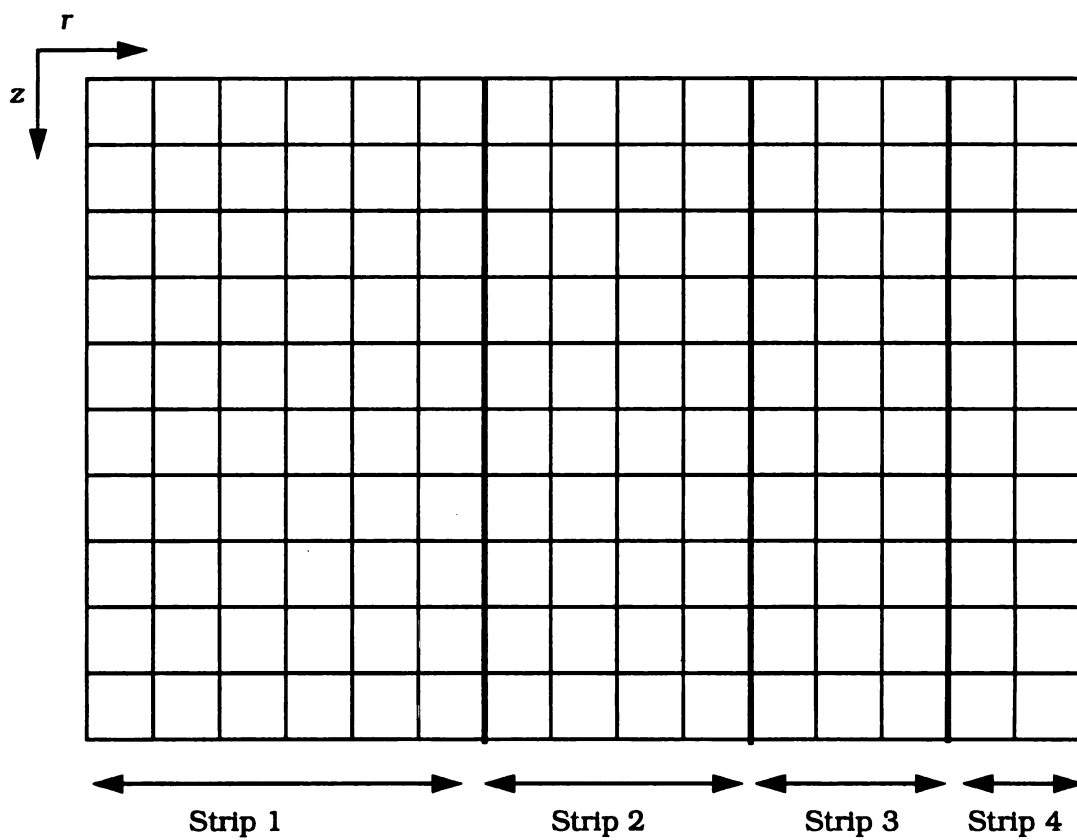


Figure 5.2: Unequal weighting implementation.

step

between

a he

follow

man

the

cre

me

mu

co

co

d

le

s

a

strip to a higher weighting strip are annihilated if the generated random number (between 0 and 1) is greater than the ratio of the two weights. The reverse case of a heavier weighted particle moving into lesser weighted zone is treated as follows. The original particle is retained with the same energy, an integral number of particles equal to the ratio of the two weights minus one is created at the same location with the same energy. The fraction left over, if any, leads to the creation of a new particle depending on the random number generated. This method has to be followed since the different weights need not be exact multiples of each other.

The unequal weighting scheme also increases substantially the number of computations required to calculate charge density. The calculation of the contribution of charge onto the background grids for each particle requires determination of the weighting factor corresponding to the region of particle location. The calculation of average particle energy is also more complicated since the total average energy is now a weighted average of the individual energy and particle weight and is given by

$$E_{avg} = \frac{\sum_i \left(\sum_j E(j) \right) W(i)}{\sum_i \left(\sum_j N(j) \right) W(i)} \quad (5.4)$$

where $W(i)$ is the weight of the *superparticle* in strip i , $E(j)$ is the energy of the particle in the strip i , $N(j)$ is the number of particles in strip i and E_{avg} is the weighted average energy of the particles.

5.2.2 Poisson's Equation Solution

The solution to Poisson's equation is done in two dimensional rz coordinate system similar to the method detailed in Section 4.2.2. As can be seen in Figure 5.1, all external boundaries are considered grounded and hence held at constant zero potential. However, the plasma source region corresponding to the area under the quartz dome, cannot be treated in this fashion. The electrons of the freshly generated electron-ion pairs move away from the ions due to their higher energy. This leads to an ambipolar field being built which attempts to slow down the electrons and speed up the ions. The Poisson's equation solver should be able to correctly resolve this field.

In order to correctly simulate this effect, the grid points in the source region were considered to follow the *Von Neumann* boundary condition wherein the derivative of the potential in the z direction is considered zero. This is done by assuming a pseudo-grid point outside the simulation region as shown in Figure 5.3. The part of Poisson's equation pertaining to the z dimension becomes

$$\frac{\partial^2 V}{\partial z^2} = \frac{\left(\frac{\partial v}{\partial z}\right)_{1,1} - \left(\frac{\partial v}{\partial z}\right)_{1,-1}}{2\Delta z} \quad (5.5)$$

where Δz is the grid spacing in the z direction, $\left(\frac{\partial v}{\partial z}\right)_{1,1}$ is the variation of the voltage at the higher z location and $\left(\frac{\partial v}{\partial z}\right)_{1,-1}$ is the variation of the voltage at a at the pseudo grid point. After applying the *Von Neumann* boundary condition

$$\left(\frac{\partial v}{\partial z}\right)_{1,1} = -\left(\frac{\partial v}{\partial z}\right)_{1,-1}, \quad (5.6)$$

equation (5.5) can be finally written in terms of internal points as

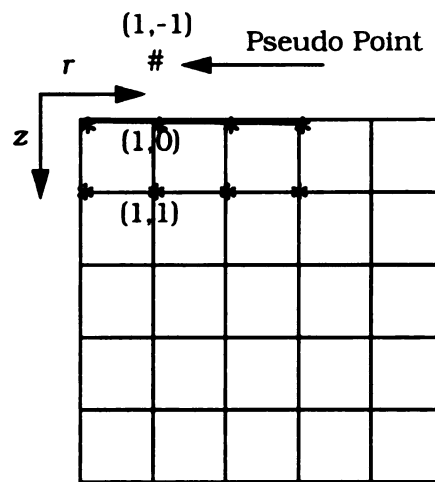


Figure 5.3: Implementation of Von Neumann condition.

$$\frac{\partial^2 V}{\partial z^2} = \frac{V(1, 1) - V(1, 0)}{\Delta z^2} \quad (5.7)$$

where $V(1,1)$ is the potential at grid $(1,1)$ and $V(1,0)$ is the potential at the desired location in the source region.

These modifications to the Poisson's equation solver in cylindrical coordinates increase the numerical complexity of the program. Further, the grid point at the origin is a special case since it is part of the central charge column and the source region. Profiles of the code running this simulation show that up to 60% of the time spent per loop is spent in the Poisson's equation solver.

Charge densities in the region of the source can be substantially higher than the average charge density of the structure. The Debye length condition has to be satisfied both globally and locally for overall stability. Therefore, the number of grids per unit length in the r direction under the quartz dome has to be more than those required for the rest of the volume. The increased number of grids also increases the time required to solve Poisson's equation.

5.2.3 Example Simulation

The 2D3V simulation system developed in this research was applied to an example system of cavity radius 9 mm, length 9 mm and quartz dome radius of 3 mm. The simulations were conducted assuming an electron temperature of 7 eV and charge density of $1.0 \times 10^9 / \text{cm}^3$.

The time evolution of particle density is shown in Figure 5.4. It can be seen that after a certain time, the simulation comes to a steady state with the rate of loss of particles being compensated for by the rate of gain from the source area.

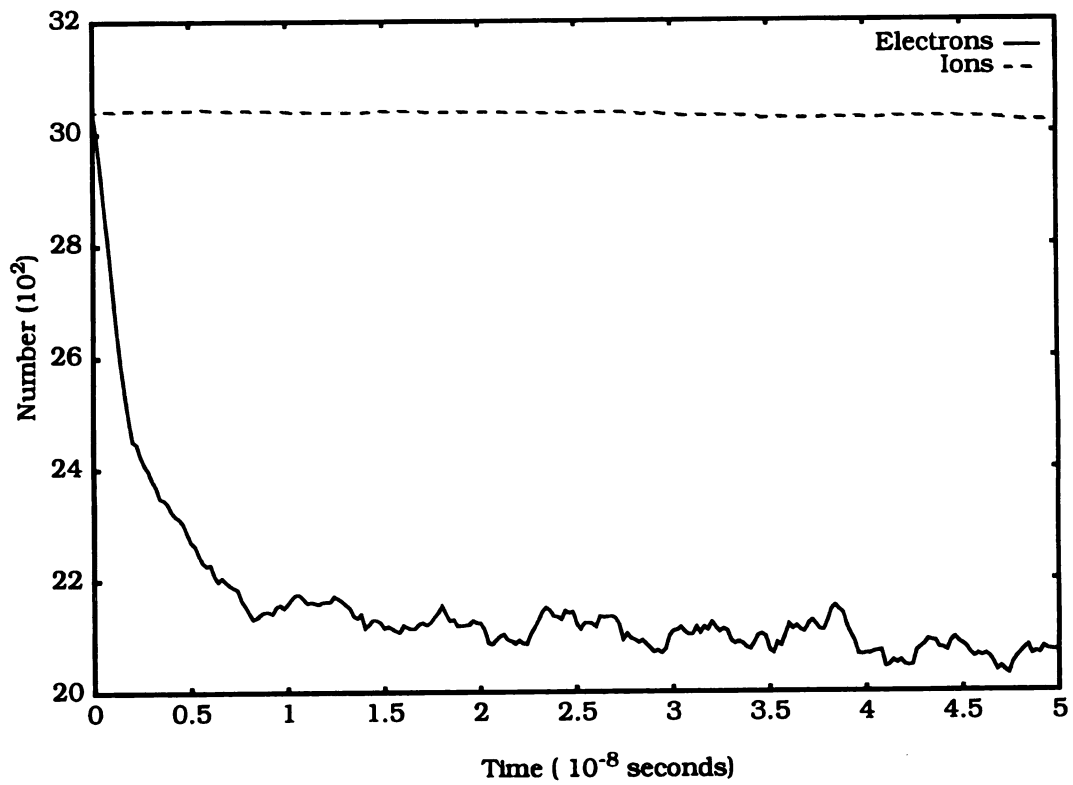


Figure 5.4: Time evolution of number of particles.

The plasma potential in the simulation area is shown in Figure 5.5 and it can be seen that the sheath around the grounded walls and the potential at the source region are clearly resolved. The time evolution of energy of the particles can be seen in Figure 5.6. From the figure it can be observed that the energy of the electrons drops rapidly and then reaches a steady state after some time. The spatial variation of electron density vs. z at a number of different r points can be seen in Figure 5.7. The electron density is higher in the source region, constant in the middle and drops off at the grounded edges. The variation of electron density vs. r for a number of different z points is shown in Figure 5.8. The variation of ion density vs. r and z is shown in Figures 5.9 and 5.10, respectively. It can be seen that the charge densities in the grids near the quartz dome are higher than the densities in the rest of the simulation area.

The simulations can also be performed assuming non-uniform source region flux. The results of electron charge density vs. z for a case where the input flux density showed a Bessel function dependence as seen in Figure 5.11, is shown in Figure 5.12. It can be seen that the charge density follows the flux density variation in the source region and is highest in the center reducing towards the edge of the generation zone. The selection of the Bessel function dependence was arbitrary, any source region spatial variation could be the input to this downstream simulation model.

The simulations presented in this chapter demonstrate the application of numerical modeling techniques to plasma processing. The example developed in this chapter was limited to a small simulation region size due to memory limitations on the single CPU workstation used. The techniques described later

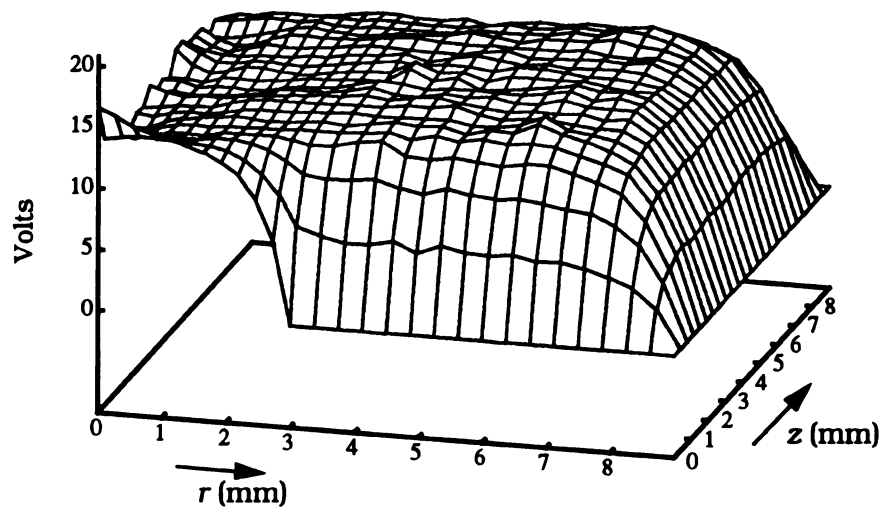


Figure 5.5: Plasma potential in the simulation region.

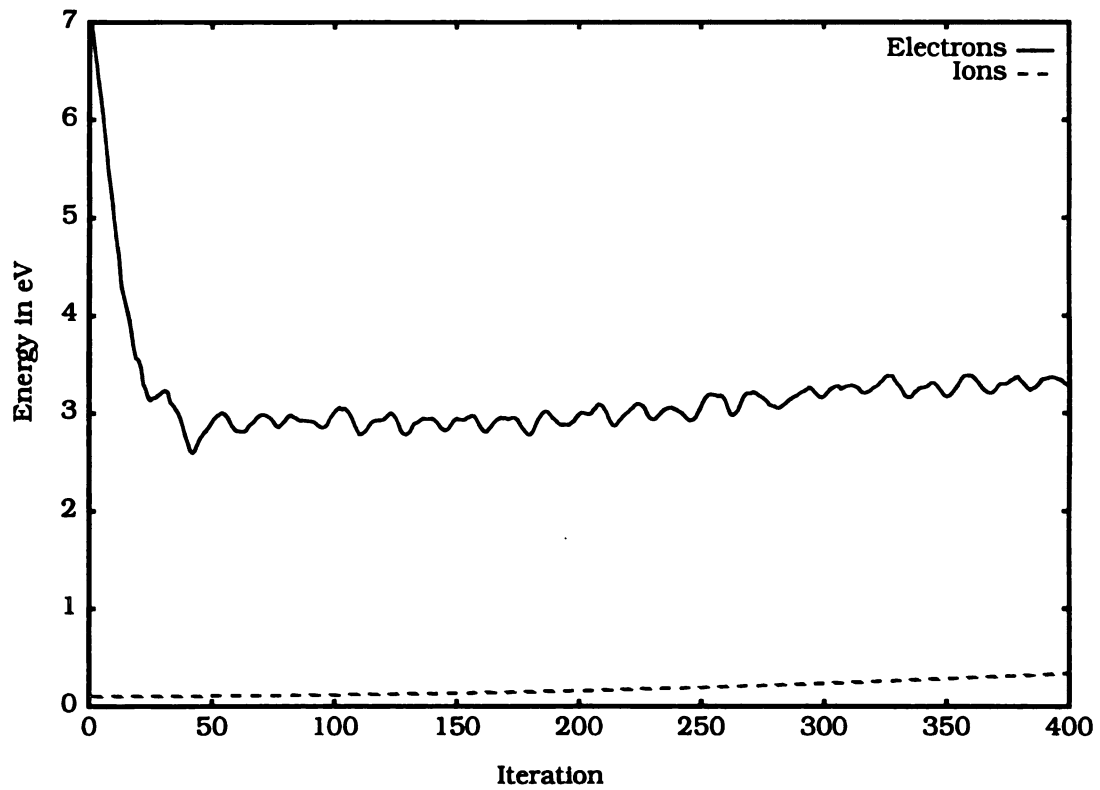


Figure 5.6: Time evolution of particle energy.

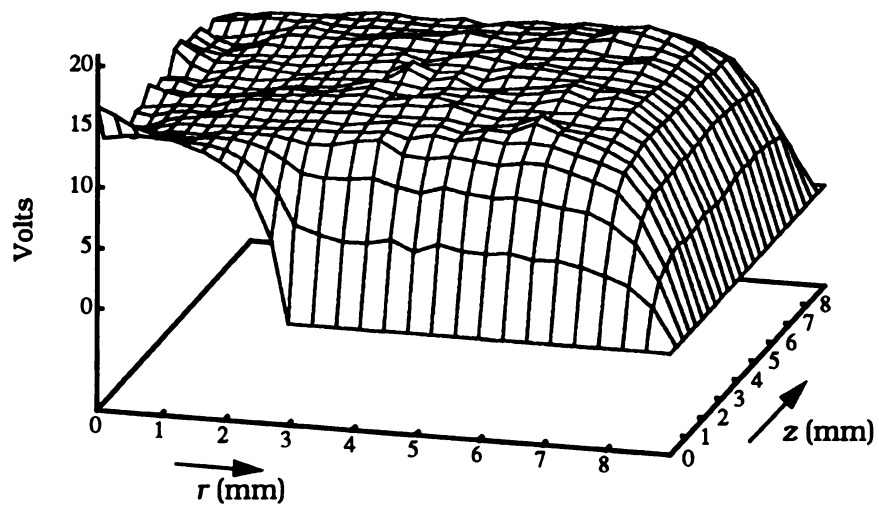


Figure 5.5: Plasma potential in the simulation region.

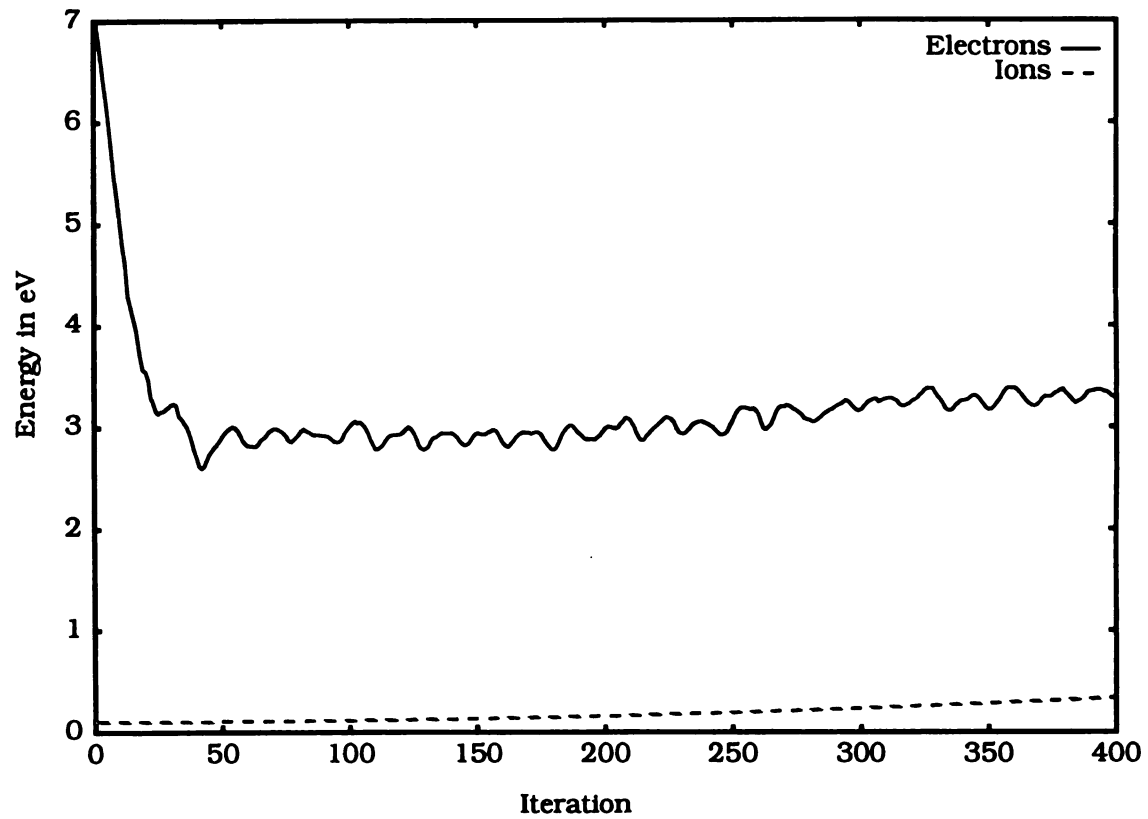


Figure 5.6: Time evolution of particle energy.

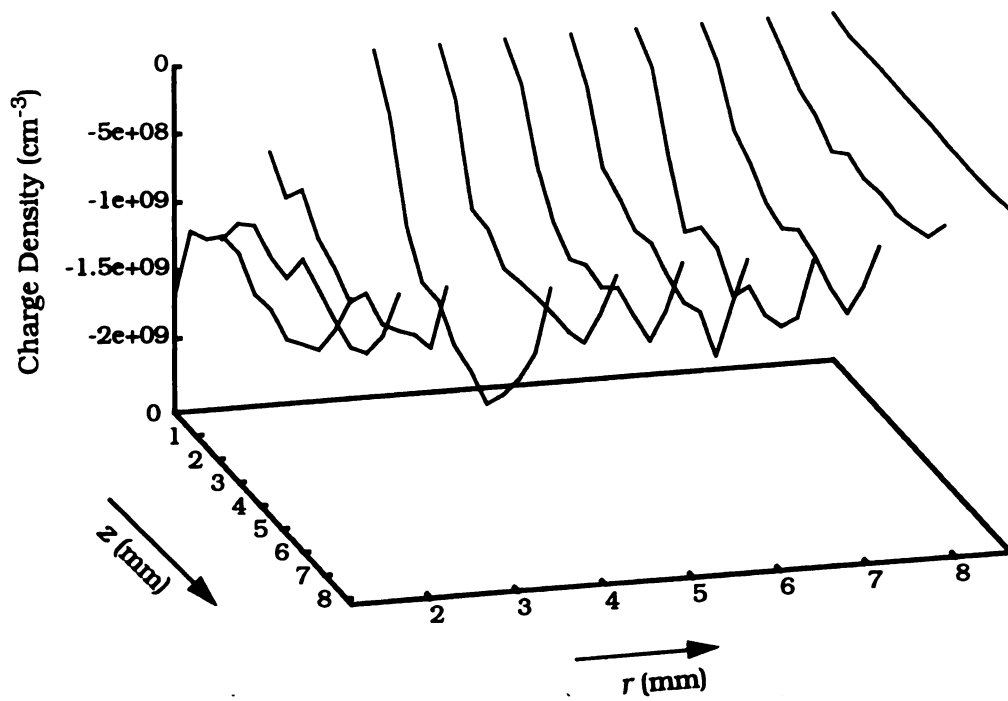


Figure 5.7: Electron density vs. z .

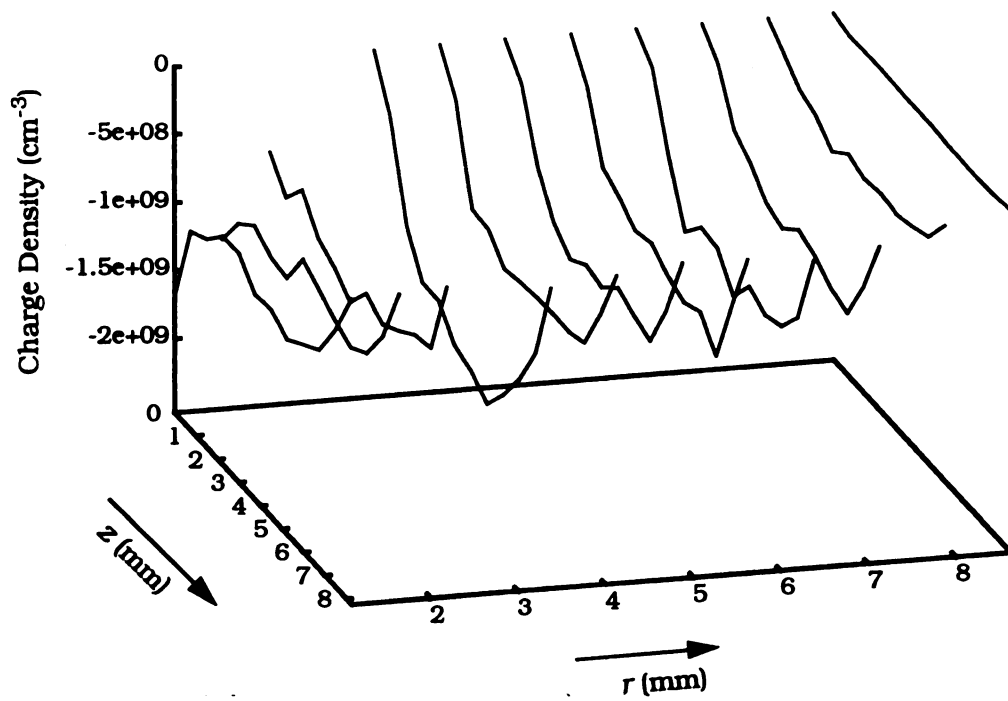


Figure 5.7: Electron density vs. z .

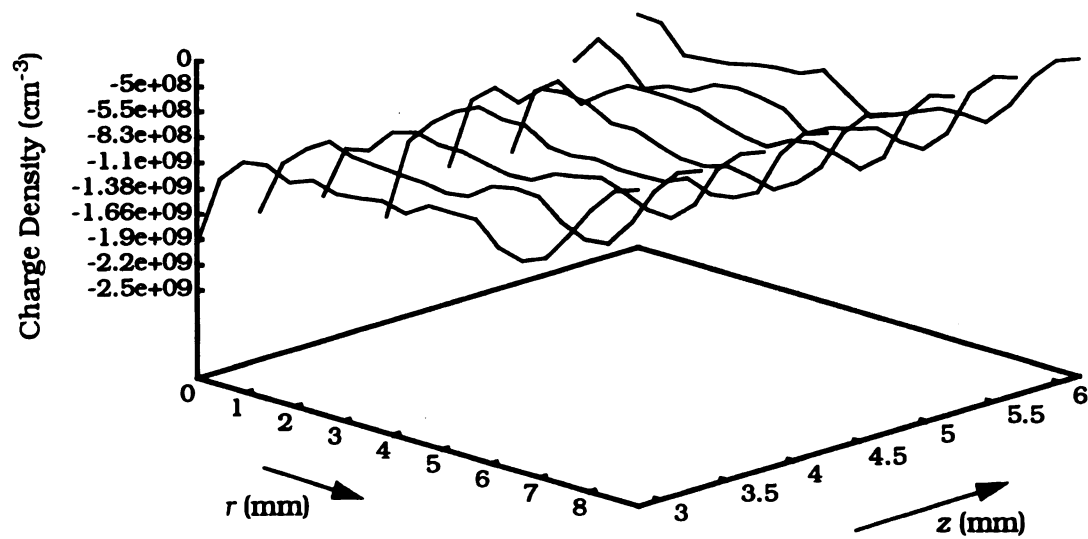


Figure 5.8: Electron density vs. r .

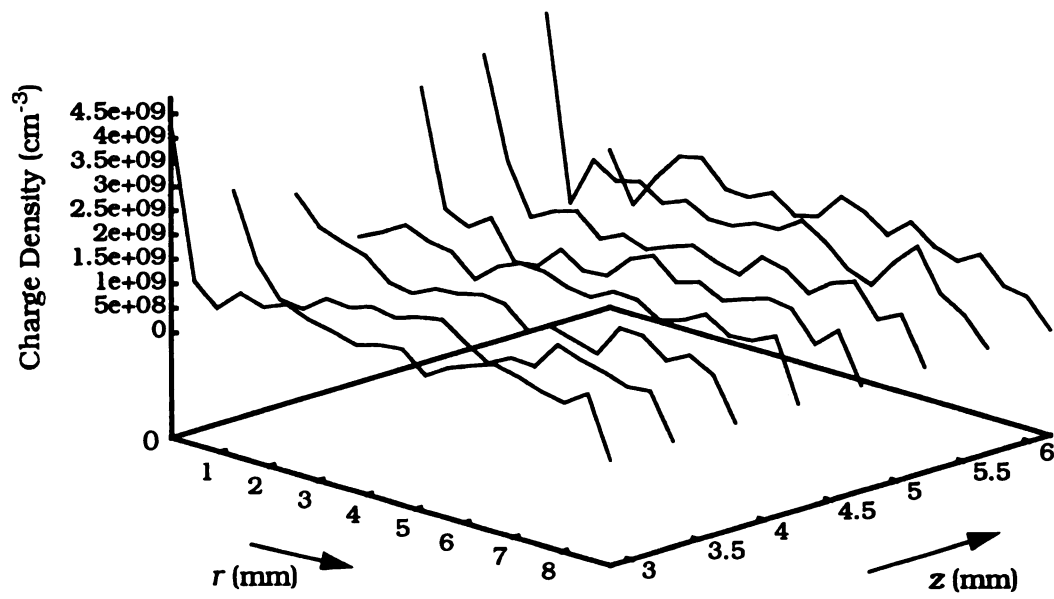


Figure 5.9: Ion density vs. r .

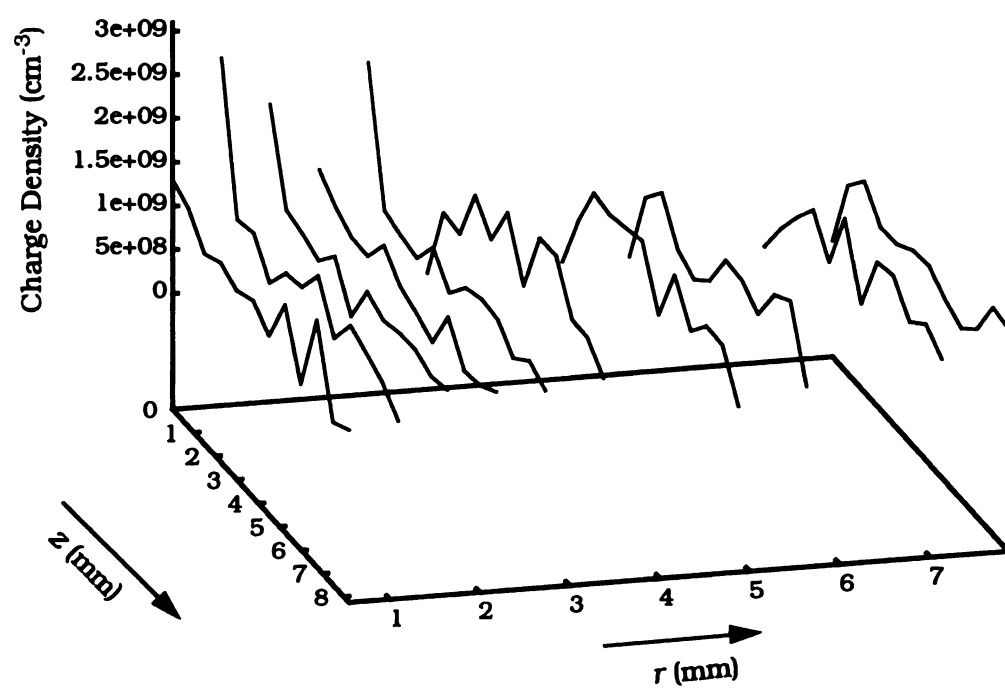


Figure 5.10: Ion density vs. z .

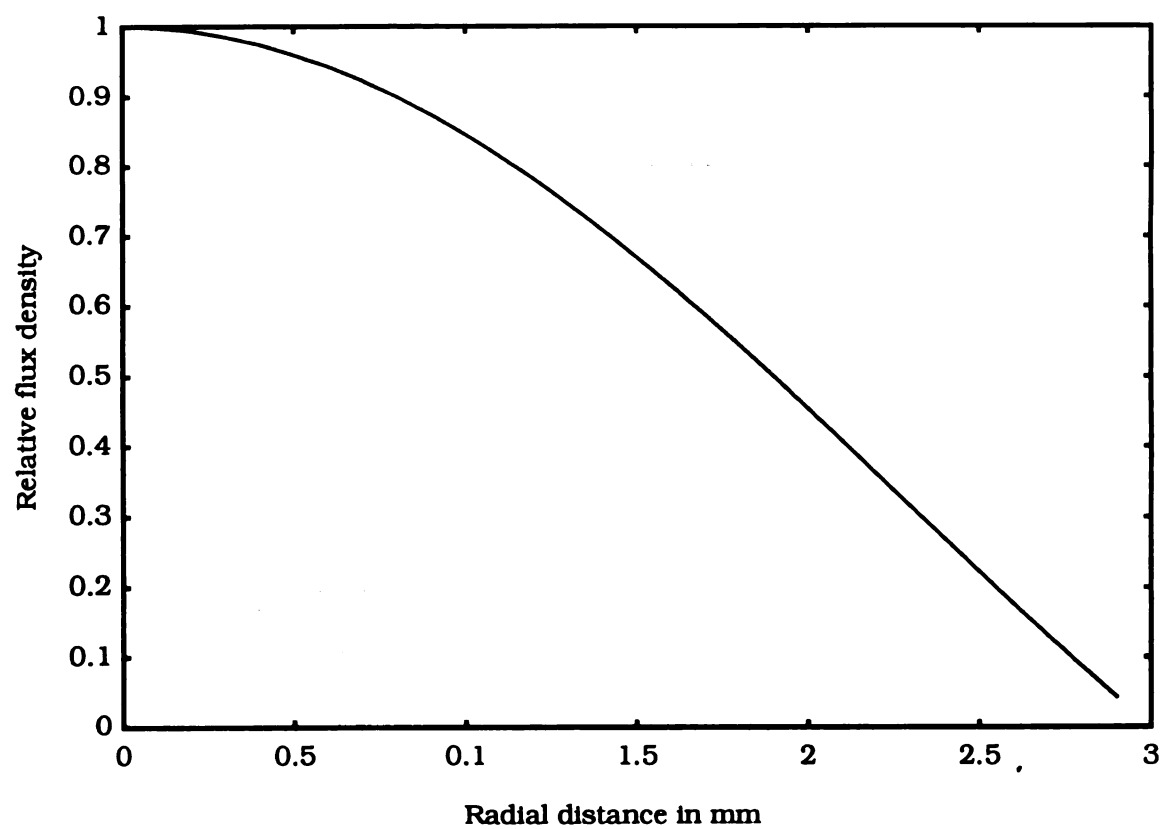


Figure 5.11: A Bessel function type flux distribution.

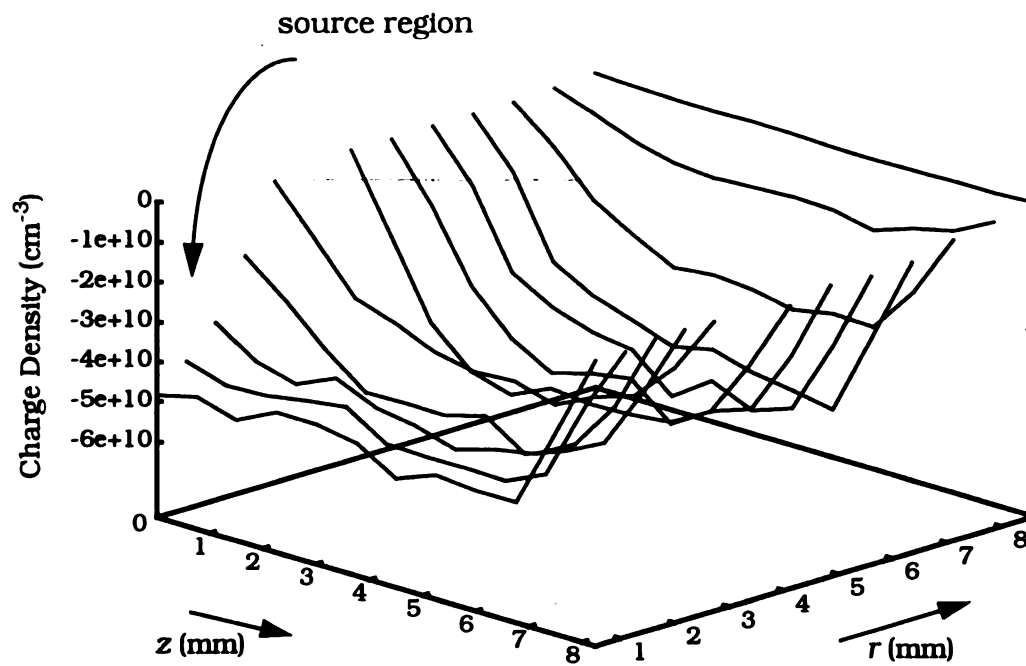


Figure 5.12: Electron density vs. z with non-uniform source.

in Chapters 7 and 8 allow the methods of this chapter to be extended to larger geometry structures.

Chapter 6

Simulation of a Compact ECR Source

6.1 Simulation of a Compact Ion Source

An example of doing a 3D3V self-consistent electromagnetic field-plasma simulation is presented in this chapter. A compact ion source which was described earlier in Section 3.2 is simulated. The permanent magnet static \mathbf{B} field in the plasma region is obtained by solving equation (2.3) beforehand [8]. Figure 6.1 shows a contour plot of the magnetic field lines in the rz plane. Of particular interest is the 875 Gauss field line around which ECR power coupling to the electrons is expected to occur. For this particular magnet configuration, the 875 Gauss line is located close to the edge of the quartz.

The structure being simulated is shown in detail in Figure 6.2. The axial coordinate of the cavity program is denoted z' and the axial coordinate of the plasma program is denoted z as shown in the figure. The plasma source geometry and operating conditions are selected to permit the simulation to run on a single CPU workstation. Simulations of an argon discharge were conducted for this source assuming an initial background charge density of $1.0 \times 10^9 / \text{cm}^3$ at a pressure of 1.0 mTorr and electron temperature of 5 eV. Under these conditions the Debye length is 5.25×10^{-2} cm. Hence, to simulate the plasma generation region, 35 grids in the radial direction and 59 grids in the axial

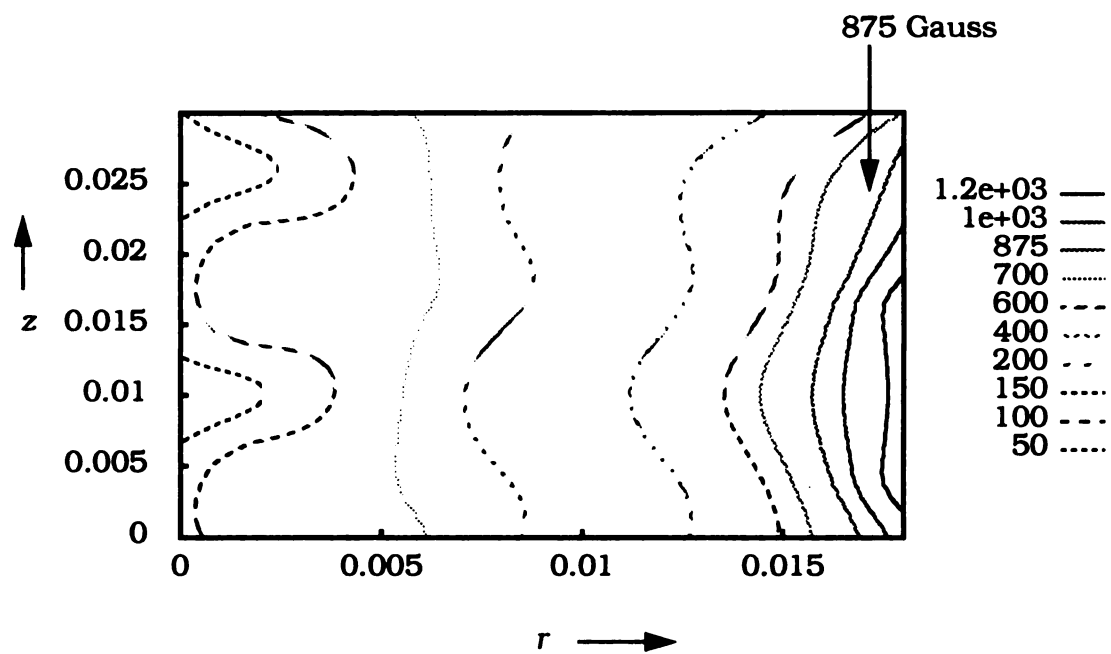


Figure 6.1: B field contours inside ion source.

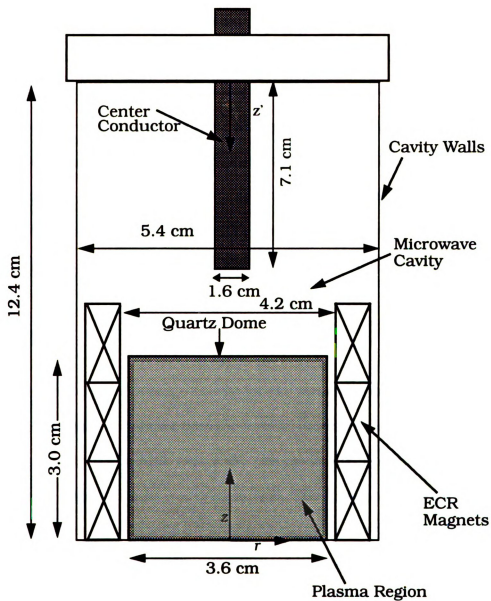


Figure 6.2: Structure being simulated.

direction are required. A non-terminated grid for a 45 degree slice in the ϕ direction was setup.

The grid structure for the EM program in the plasma region had 18 grids in the radial direction, 30 grids in the axial direction and 5 grids in the ϕ direction. This leads to the EM program containing less than half as many total grid points as the plasma program for the same region and so the data acquired from the EM program was interpolated to the extra grid points in the plasma region. In order to ensure stability, the EM timestep was set at 5×10^{-14} seconds and the plasma timestep was set at 2×10^{-11} seconds. This leads to communication between the two programs every 400 iterations as shown in Figure 4.8. Simulations on a HP 735 took an average of three days to reach steady state.

6.2 Results

The time evolution of average energy for the structure is given in Figure 6.3. This example simulation shows that the structure settles down to a steady state electron temperature of a little over 4 eV. The power absorption of the structure can be seen in Figure 6.4 as a greyscale map. The location of maximum power absorption is given by the whitest area and can be seen to correspond to the location of the 875 Gauss line. The simulation predicts that even though the 875 Gauss line is present all along the quartz, the conditions favorable for ECR heating, where the \mathbf{E} field is perpendicular to the \mathbf{B} field, exist only in the region closest to the microwave input. The time evolution of particle numbers is given in Figure 6.5. It can be seen that there is an initial outflow of the more energetic electrons but later, the combination of built up plasma sheaths and ionization

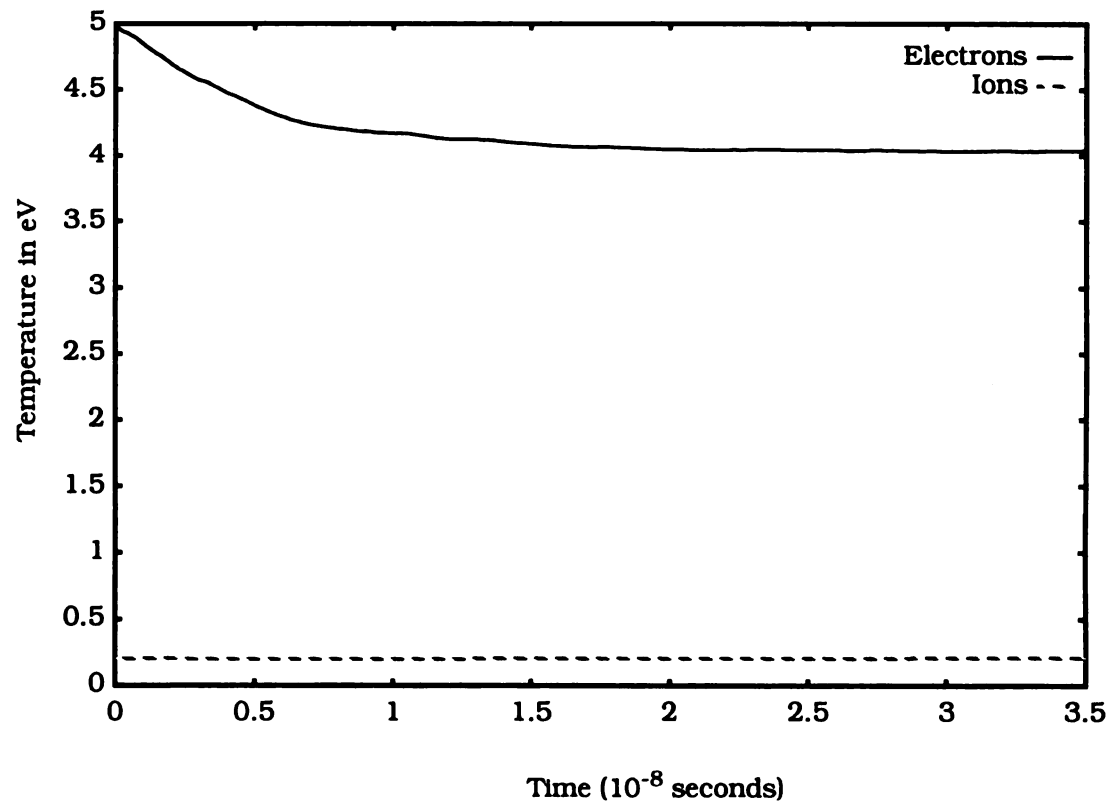


Figure 6.3: Time evolution of average particle energy.

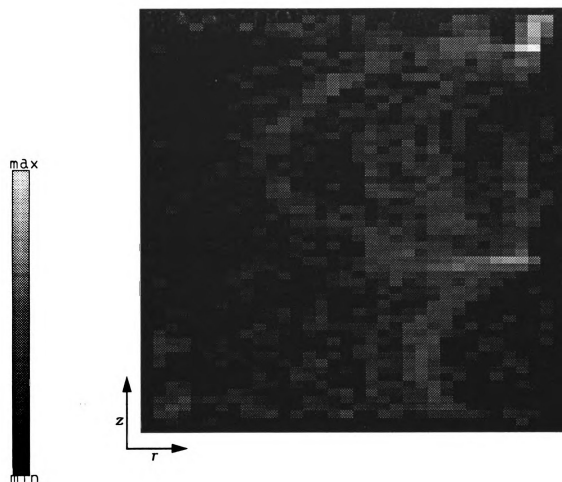


Figure 6.4: Power absorption locations

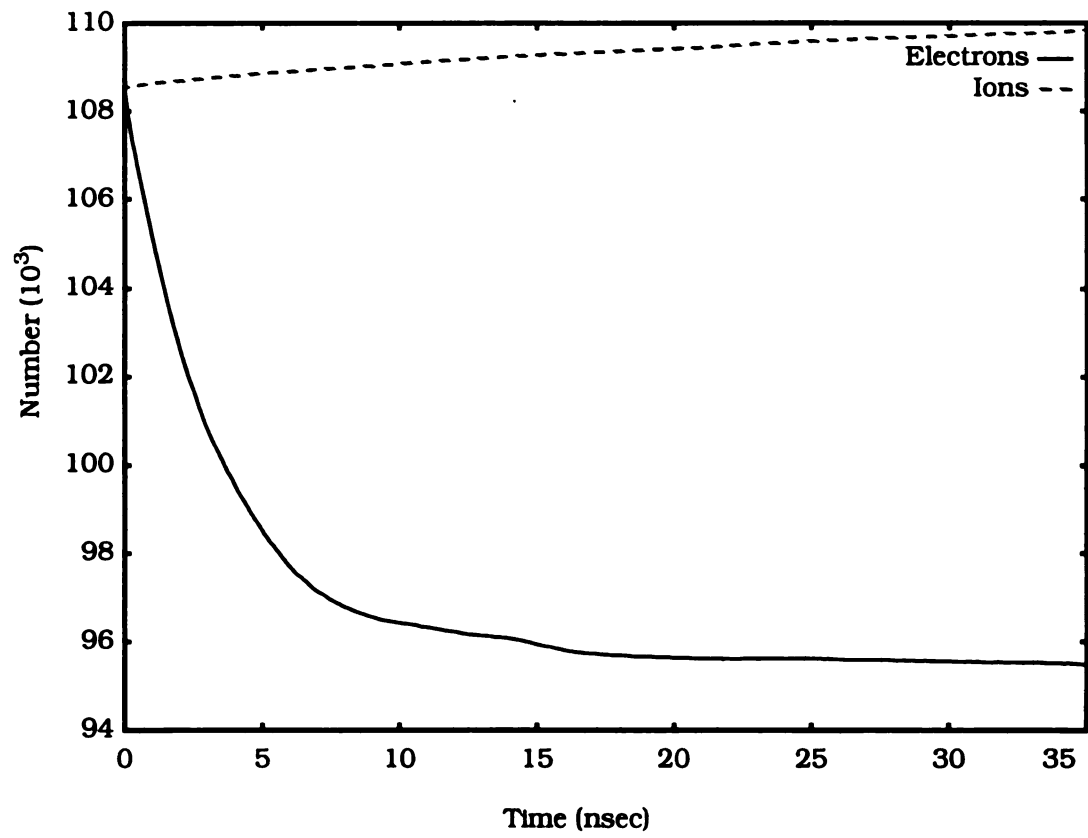


Figure 6.5: Time evolution of number of particles.

brings the number of electrons to a steady state. The plasma potential in the simulation structure can be seen in Figure 6.6. The time averaged values of E_r along the edge of the cavity (maximum r location) is shown in Figure 6.7. The electron energy distribution function for this example is given in Figure 6.8. The energy shows a peak at approximately 0.5 eV and drops off rapidly at higher energies. The self-consistent electromagnetic field-plasma simulation model calculates several important plasma source characteristics including plasma potential, plasma sheath thickness, electromagnetic field strengths, microwave power absorption characteristics, and particle energies. The general techniques of doing the EM-plasma simulation developed in Chapter 4 have demonstrated themselves for the compact ion source simulated in this chapter. The next two chapters present techniques that allow the extension of these techniques to larger plasma sources and higher plasma densities where larger grid structures and more *superparticles* are needed. These extensions involve the use of advanced computing resources.

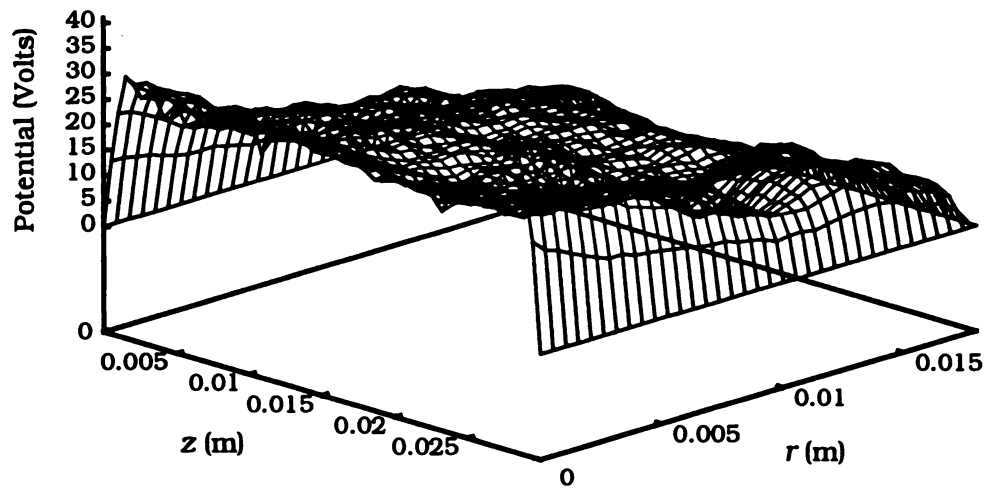


Figure 6.6: Plasma potential in the rz plane.

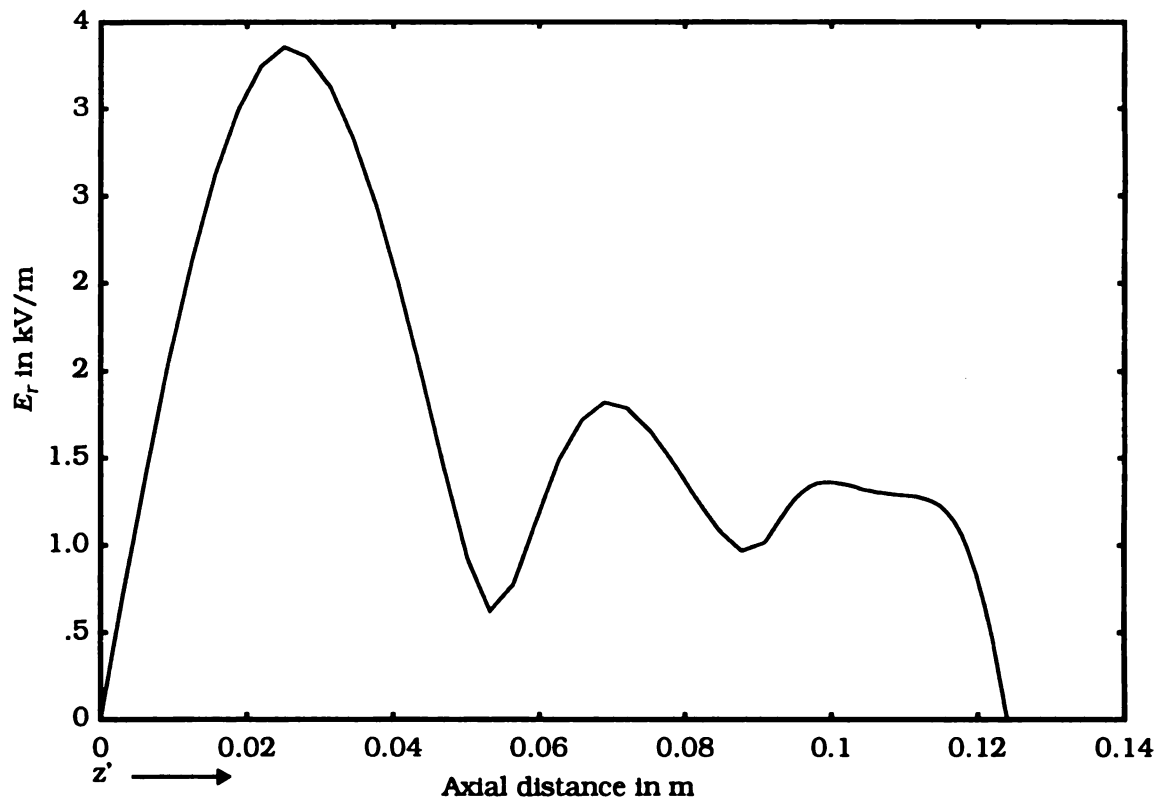


Figure 6.7: Time average values of E_r along the edge.

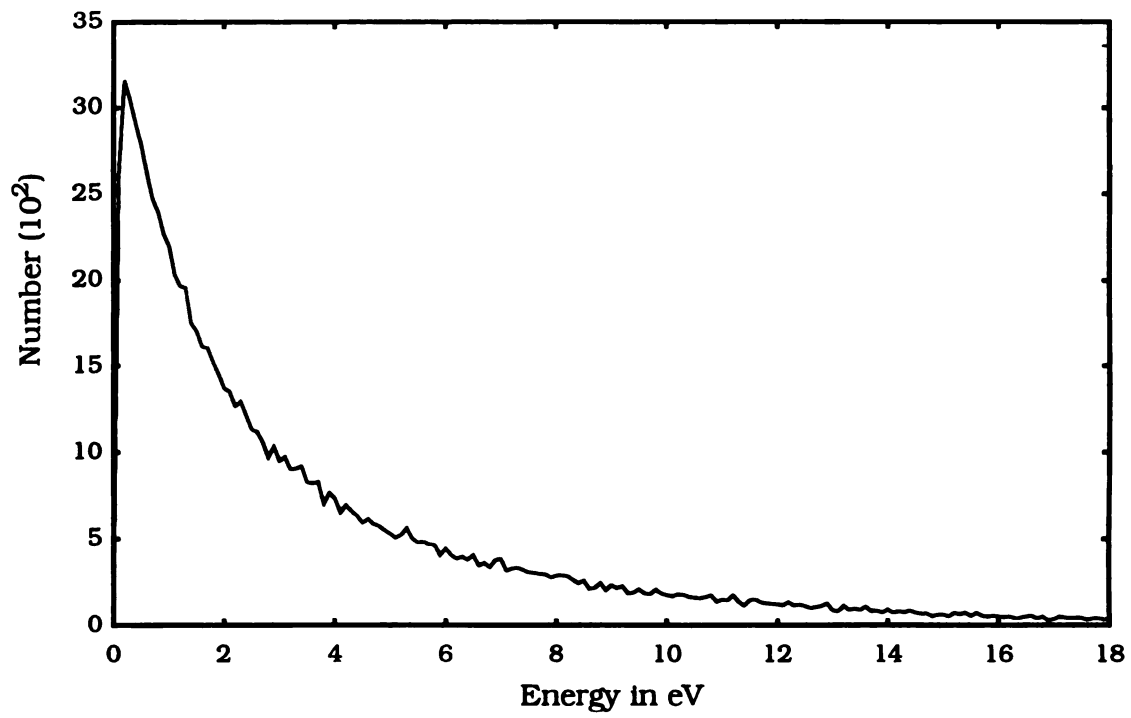


Figure 6.8: Electron energy distribution function.

Chapter 7

Implementation on the MasPar MP2 System

7.1 Introduction

The movement of particles in particle type plasma simulations are essentially independent of each other once the governing forces are resolved. This gives rise to the possibility of implementing a simulation wherein the motion of each particle is resolved and computed independent of the others. Potentially, this type of simulation can give considerable speed-up over serial implementations. In this research, a Massively Parallel Processor (MPP) containing a number of Processing Elements (PEs) has been used to implement this idea. Each PE has a certain volume assigned to it and performs computations on the particles that lie in this volume.

The system chosen to implement a parallel version of a plasma simulation was the MasParTM MP2 system. Systems of the type MP2 are termed as data parallel or SIMD (single instruction multiple data) machines. In SIMD machines, several relatively small processors execute the same program instruction simultaneously. Massively parallel SIMD units consist of one instruction fetch unit and at least 1000 data processors. The instruction stream is processed serially by a control unit and dispatched to the processing elements (PEs) that execute them in parallel. These are also called data parallel since the only

distinguishing feature of each processor is the data it operates upon. A typical MP2 consists of a front end and a data parallel unit (DPU).

The front end is a modified workstation that runs a flavor of the popular UNIX operating system. The DPU consists of an Array Control Unit (ACU), an array of at least 1024 PEs and communication mechanisms. The ACU is a processor with its own registers and data and instruction memory. Certain types of serial instructions can be executed on the ACU. The ACU can be considered as the slave master which drives the PEs in lock step and sends data and instructions to each PE simultaneously.

The DPU is where all the parallel processing is done. Each PE consists of a 32 bit load/store arithmetic processing element with dedicated registers and 64 KBytes of local RAM. Each RISC (Reduced Instruction Set Computer) PE unit is rated at 4.15 MIPS and 0.38 MFLOPS [82]. There is no floating point chip in a MasPar system, though there is some special hardware to improve floating point capability. The floating point operations are microcoded and hence take several clock cycles. SPEC benchmarks are not strictly applicable to parallel systems and are hence unavailable on the MP2. These PEs are distributed physically as a two dimensional array which can be 1K-16K in size and are scalable between the limits in powers of two. Each non-overlapping square matrix of 16 PEs in a PE array is termed as a PE cluster. For example, a 1K PE array is made up of 64 4x4 PE clusters.

One of the major aspects, and a source of bottleneck, of a massively parallel system is its communication subsystem. Communication between PEs and the ACU occurs over a special bus. Communication between two PEs consists of Xnet and global router [83] communications.

Direct communication by any PE to any other PE which is in a straight line in the eight directions, i.e., North, South, East, West, North-East, North-West, South-East and South-West, can be achieved by a special communication protocol called Xnet. Xnet can provide scalable performance of up to 23 GBytes/sec with variations that can provide wrap around on edges and pipelined versions for long distance communication.

The router communication system provides a method of communication between any PE to any other PE which may not necessarily satisfy the Xnet criterion. This form of communication utilizes a three-stage, circuit switched crossbar with a peak bandwidth of 1.3 GBytes/sec.

The target system available at the Scalable Computing Laboratory at Ames Laboratory at Iowa State University consists of a 64x64 MP2 array with each PE having 64 KBytes of local RAM. The front end is a DEC workstation running ULTRIX operating system.

In addition to all of the hardware, the MasPar systems have software programming and debugging tools. The language used to implement the parallel version of the application was the Maspar Programming Language (MPL). This language is an extension of standard ANSI C with constructs to provide for serial and distributed variables. Debugging support is provided by a visual debugger, *MPPE*, [84] which allows setting break points and viewing the data and state of processors in the PE array. The data can be viewed either as numbers or graphically visualized as a 2D graph with the points representing the value of the variable being probed. Further, graphical routines are available which can be used to build graphical display routines for the required data. Experiences with visualization tools are given in Section 7.5.

7.2 Implementation of a Two Dimensional Parallel Program

The processors on the Maspar MP2 are arranged in the form of a two dimensional array which leads to the possibility of a direct implementation of a two dimensional particle-type plasma simulation. With this in mind, a program simulating a 2D3V (two spatial dimensions and three velocity dimensions) plasma particle simulation on a 32x32 grid was directly implemented on the MP2 by considering each PE as representing one grid point. The MasPar command *mpswopt* [85] was used to make the final program run on a 32x32 portion of the PE array. The PEs present on the edges were considered as boundary elements and were assigned grounded boundary values for the potential. Each PE was assigned eight electrons and ions initially. The initial locations of these particles were randomly distributed in the PE's domain and their initial energies were distributed according to a Maxwellian distribution across all the processing elements. Some of the major issues involved in parallelizing this application involved the movement of particles which leave a PE's domain, solving Poisson's equation and the calculation of the local electric and magnetic fields, all of which require extensive use of communications.

7.2.1 Movement of Particles Across PEs

A particle leaving the PE is removed from the current location and moved to the correct PE so as to preserve the total particle count. This involved locating the direction of particle motion in terms of the eight directions and transferring the total details of the particle, including its location in space and velocity to that neighbor using Xnet. The particles that left the boundary PEs beyond the simulation region were considered lost to the grounded walls.

7.2.2 Poisson's Equation Solution

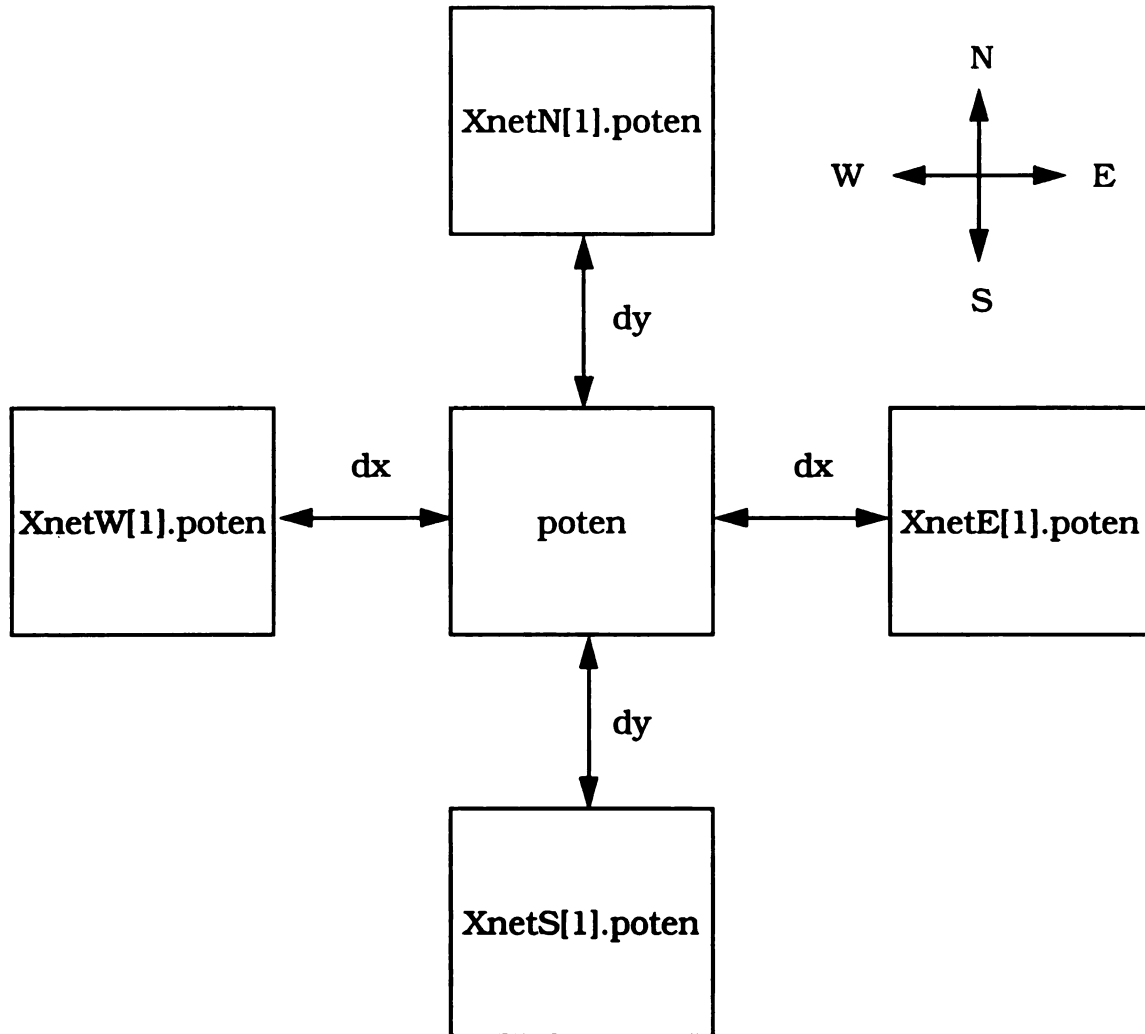
The solution of Poisson's equation was done very simply by using the potential values from the current PE's north, south, east and west neighbors as shown in Figure 7.1. The technique used to solve Poisson's equation was the Gauss-Seidel iterative method applied to a finite difference formulation. In this case, since the communication is always with an immediate neighbor, Xnet communications were used to enormous speed advantage.

7.2.3 Calculation of Fields

Local electric fields at each PE are calculated by using the potential values, found from solving Poisson's equation, at its immediate neighbors. For example, E_x , the electric field in the x direction is calculated using the difference in the value of potential from the eastern and western neighbors. Similarly the y component of the field is calculated using the northern and southern neighbors as shown in Figure 7.2. The problem of calculating the value of electric and magnetic fields acting upon a particle involves computing the contribution of the electric fields from the neighboring four PEs as shown in Figure 7.3. Therefore, significant communication overhead can be involved in this part of the program.

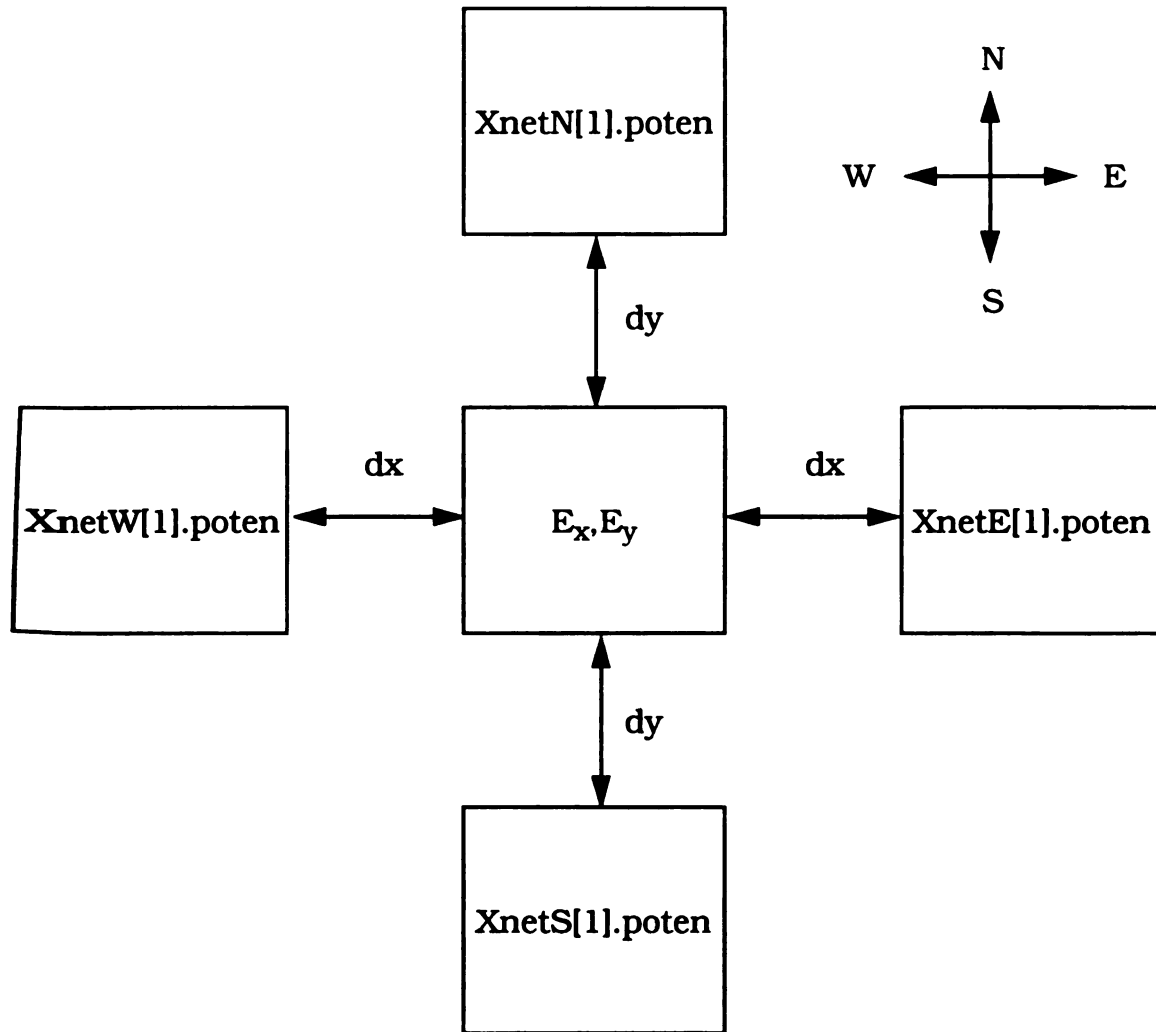
7.2.4 Results of the Two Dimensional Simulation

Implementation of a two dimensional program was quite successful and the program was not communication bound as was initially feared. Profiling the parallel code showed that the PEs spent most of the time in moving the particles (solving the Lorentz force equation) than in communicating any of the variables



If ($dx = dy$) then $poten = (XnetN[1].poten + XnetS[1].poten + XnetE[1].poten + XnetW[1].poten)/4$

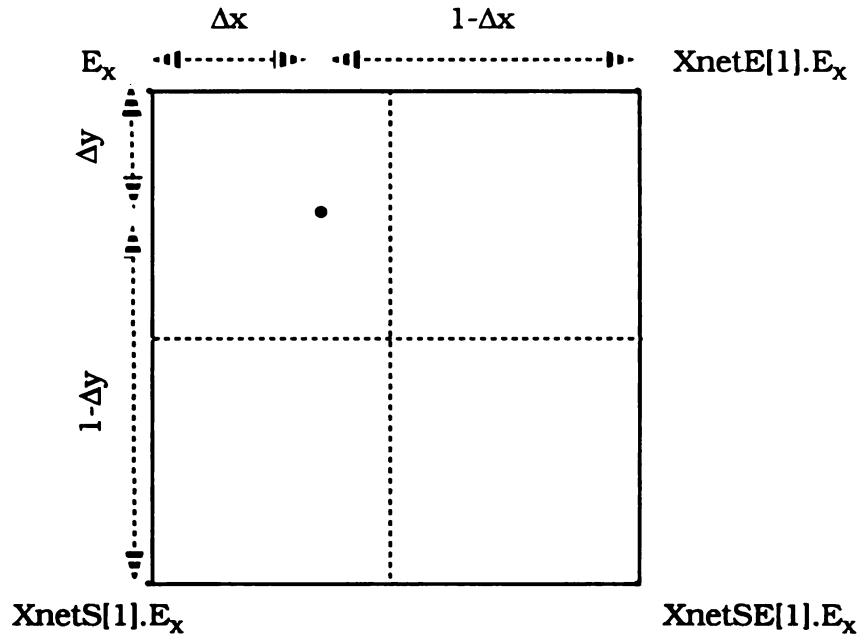
Figure 7.1: Solving Poisson's equation.



$$E_y = (X_{netN}[1].poten - X_{netS}[1].poten) / (2 * dy)$$

$$E_x = (X_{netW}[1].poten - X_{netE}[1].poten) / (2 * dx)$$

Figure 7.2: Solving for electric field.



$$\begin{aligned}
 E_x^\bullet &= (E_x * (1-\Delta x) * (1-\Delta y)) \\
 &+ (XnetE[1].E_x * (\Delta x) * (1-\Delta y)) \\
 &+ (XnetSE[1].E_x * (\Delta x) * (\Delta y)) \\
 &+ (XnetS[1].E_x * (1-\Delta x) * (\Delta y))
 \end{aligned}$$

Figure 7.3: Interpolation of x component of \mathbf{E} -Field at particle location.

discussed earlier. Table 7.1 tabulates the specifications of the three platforms [86], [87] and Table 7.2 tabulates the time taken per iteration on these three platforms for a 32x32 grid.

It can be seen that the MP2 was not much faster than the HP 735 and over three times as fast as the Sparc 10. The next test conducted was to check the scalability of the problem. While scaling the problem up to a 64x64 array size did not slow the MP2 program by much, it made the HP 735 program nearly 10 times slower. This was due to the fact that increasing the number of grid points also increases the overall number of particles required for the simulation in a serial program while leaving the parallel version unaffected. These results provided encouragement to implement a complete three dimensional parallel simulation.

7.3 Implementation of a 3 Dimensional Parallel Program

7.3.1 Implementation of Third Dimension in Memory

Since the MasPar PE array is in the form of a two dimensional array, it does not lend itself in a simple way to scale a 2D program into three dimensions. One of the simplest solutions was to simulate the third dimension in the local memory of each PE. In this case the local variables such as potential and electric and magnetic field vectors are no longer single variables as in the 2D case but are vectors of length required to simulate the z dimension as shown in Figure 7.4. This approach uses a large amount of local memory since in addition to the local variable vectors, the details of many more particles have to be stored in

Table 7.1: Specification of the three systems

Platform	Clock Speed	MIPS Rating	MFLOPS Rating	SPECint92	SPECfp92
Sun Sparc 10	36 MHz	101.6	20.5	45.2	54
HP 735	99 MHz	125	45	109.0	168.0
MasPar MP2 per PE	12 MHz	4.15	0.38	NA	NA

Table 7.2: Results for the 2D simulation

Platform	Size	Time/ iteration in seconds	Speed-up Sun/HP	Normalized Speed-up	Efficiency %
Sun Sparc 10	32x32	2.18	–	–	–
HP 735	32x32	0.8662	–	–	–
MasPar MP2 (1024 PEs)	32x32	0.629	3.47/1.38	187.2/159.3	18/16
Sun Sparc 10	64x64	23.93	–	–	–
HP 735	64x64	6.9773	–	–	–
MasPar MP2 (4096 PEs)	64x64	0.69086	34.68/10.11	1871.0/1166.9	46/28

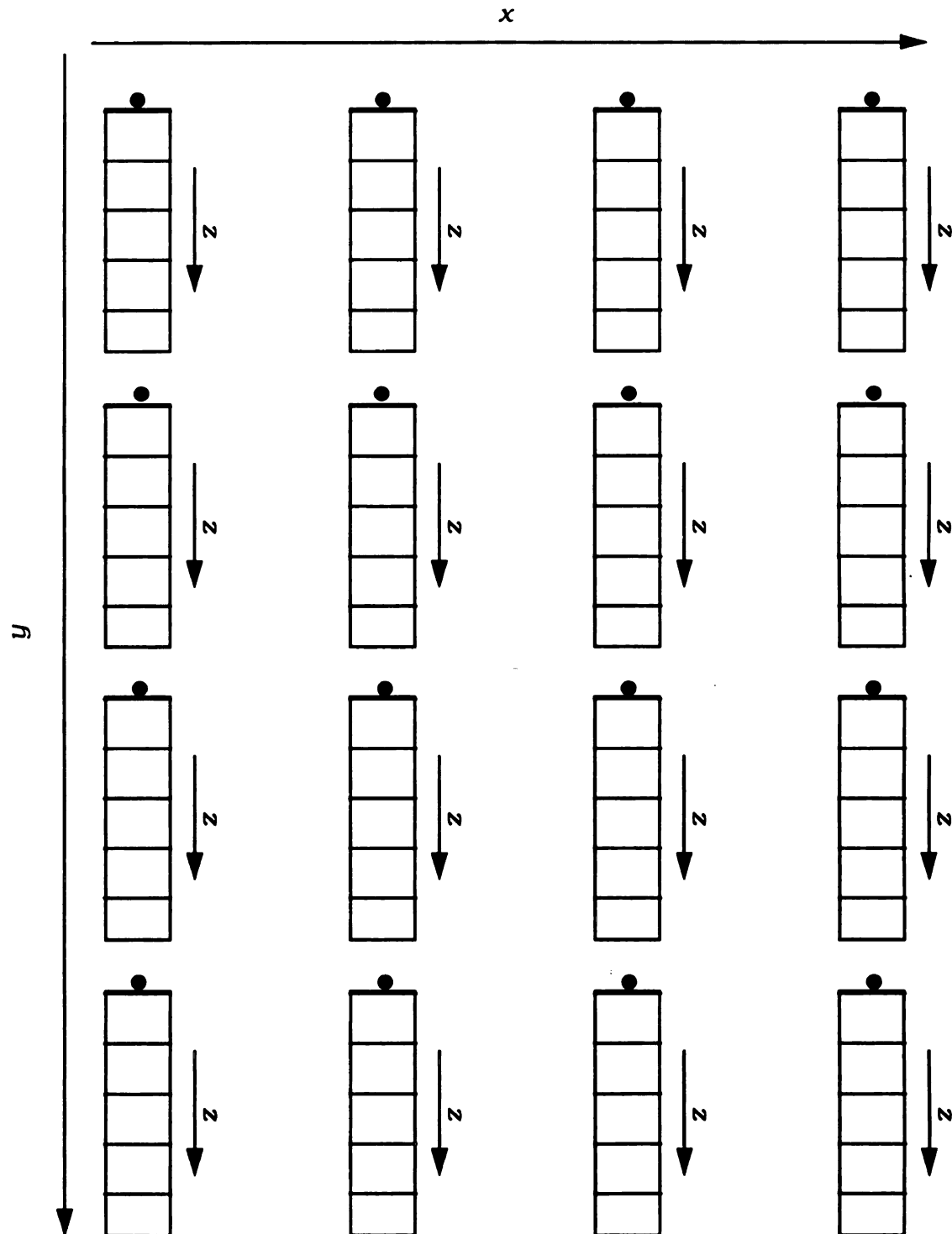


Figure 7.4: Implementation of 3rd dimension in memory.

local memory. A $32 \times 32 \times 32$ problem was implemented on a 32×32 PE array with the size of local variable vectors being 32.

The motion of particles across PE boundaries was identical to the 2D case although in this case the presence of considerably more particles per PE led to more computation and communication. The solution of Poisson's equation was similar to the 2D case but involved a loop on the z dimension and hence also led to more computation and communication. The interpolation of electric fields was found to be time consuming since the computation of the component of the fields at a particle involved twice as many communications as the 2D case. This coupled with the fact that there are more overall particles increased the time spent in communication to 8% of the total time spent per iteration, up from less than 1% in the two dimensional case. It was further noted that the system was heavily compute bound since most of the time was spent in moving the particles. In spite of this, the parallel 3D3V program was found to be nearly 4 times faster than a HP 735 serial version and nearly 13 times faster than a Sparc 10. Due to these encouraging results it was presumed that further speed-up could be obtained by spreading the load of the third dimension across processors as described in the next section.

7.3.2 Implementation of Third Dimension Across PEs

It was noticed that implementing the third dimension in the PEs local memory disadvantaged the PE array in terms of loading them down computationally. So the next design considered simulating a $32 \times 32 \times 32$ three dimensional grid across the whole 64×64 processor array. The third dimension

was simulated by dividing it across a group of four processors (called a quad), two each in the two physical directions of the PE array as shown in Figure 7.5, with 32 such groups in the x and y directions. It was theorized that the increase in the complexity and amount of communication involved would be more than offset by the increase in speed due to reduced computation overhead. This implementation involved reworking communication processes derived in the 2D and simple 3D cases.

7.3.2.1 Movement of Particles Across PEs

The PEs in each quad were numbered as an ordered pair (i,j) with $(0,0)$ simulating the top part of the z dimension and $(1,1)$ simulating the bottom part of the z dimension as shown in Figure 7.5. The movement in the z dimension when the particle leaves the 0th grid on $(0,0)$ or the last grid on $(1,1)$ is considered as loss to the grounded walls. Motion in the xy plane such that there is no z motion across processors is handled simply by moving the particle two steps in one of the eight elementary directions instead of the simple one step in the earlier cases. Further, motion of a particle in the z plane alone, not involving motion crossing xy boundaries, is handled simply by moving the particle to the correct neighbor one step away. For example, a particle moving from $(0,0)$ to $(1,0)$ in the quad would be moved one step east and one that moves from $(1,0)$ to $(0,1)$ involves a move one step south-west.

The motion of particles involving movement in which their final location demands a motion across PEs in the xy plane as well as the z plane is much more complicated. The motion of particles can be reduced to simple combination

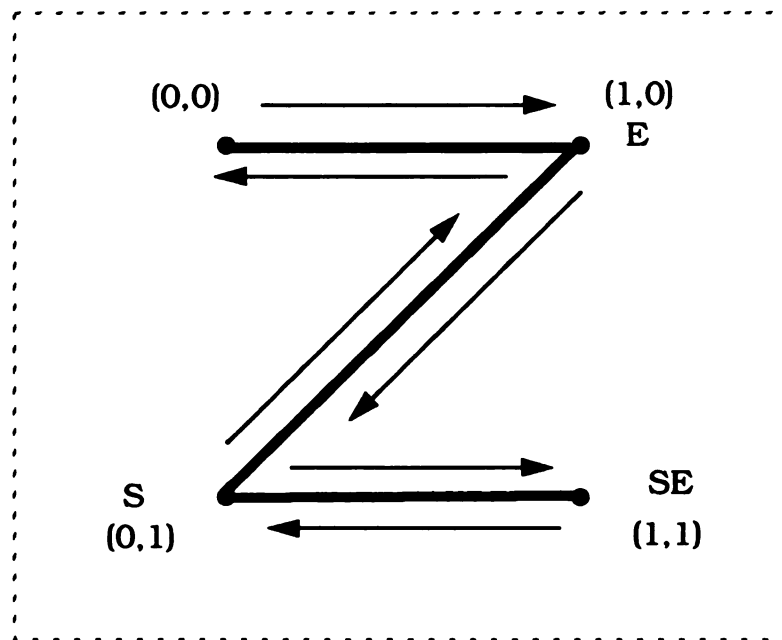


Figure 7.5: Implementation of 3rd dimension across PEs.

of directional functions. Suppose the motion in the xy plane requires the particle to move two steps NE in the simulated space, while the motion in the z plane demands that the particle be moved to a PE one step deeper (east) of PE (0,1). Since two steps in the simulated NE direction is equivalent to a step (-2,2) in the y and x directions and E is equivalent to (0,1), the final destination can be considered as NE(E) resulting in directional position of (-2,3) as can be graphically seen from Figure 7.6. The same logic can be applied to the many different combinations of xy and z motions.

7.3.2.2 Poisson's Equation Solution

The solutions to Poisson's equation is slightly more complicated than the earlier cases since calculation of potential requires communication for some points lying on the z plane also. For example, calculating the potential at the 0th grid point of the PE (1,0) in the quad would require the potential at the last grid point of (0,0). A similar situation arises at the last grid point requiring the potential at the 0th grid point of the PE (0,1). The location of the four neighbors in the xy plane is the same as in earlier cases except that they are two steps away in this case. Since there are one-fourth as many grid points at each PE as in the earlier case, there is less total communication and computation than for the earlier simpler 3D version.

7.3.2.3 Calculation of Fields

Calculation of local electric fields is similar to the earlier cases except that neighbors in the xy plane are two steps away. The calculation of fields in the z plane requires communication with PEs downstream in the z direction in a

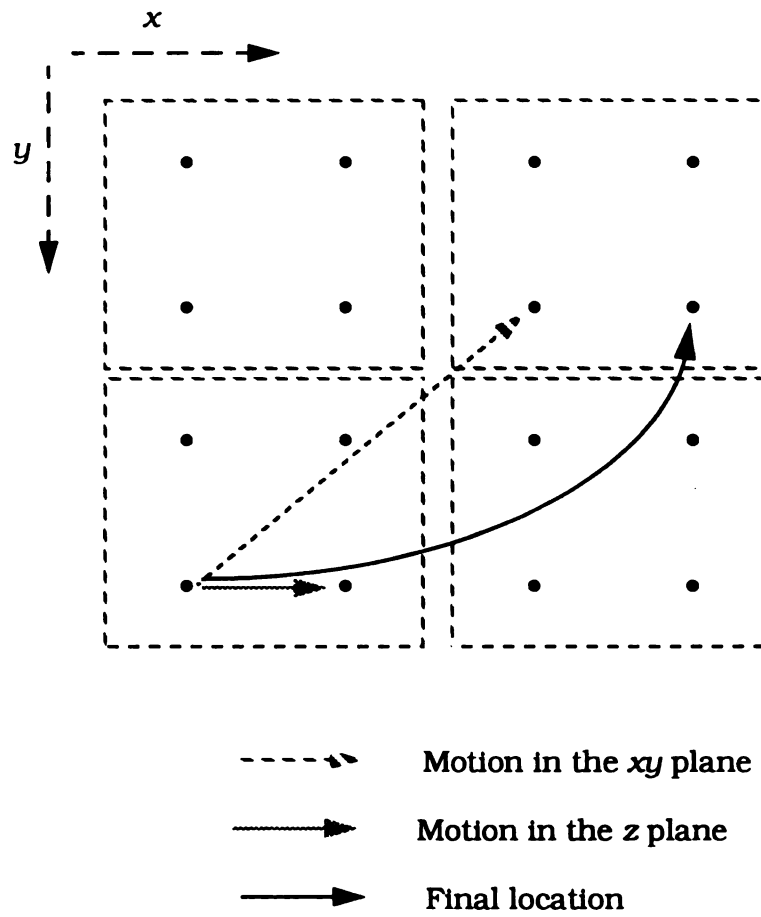


Figure 7.6: Motion of particles across PEs in the z and xy planes.

manner similar to the case discussed in Section 7.3.2.2. The interpolation of fields on to the location of a particle is also similar to the earlier cases except for the peculiarities discussed in earlier sections.

7.4 Implementation of 3D Maxwell's Equations Solution

Since a complete plasma simulation includes the coupling of Maxwell's equations with particle motion, it was decided to implement a Finite-Difference-Time-Domain (FDTD) parallel implementation of a simple rectangular microwave cavity with perfectly conducting walls to test massively parallel solutions. Equations (2.1) and (2.2) can be expanded into a set of six linear interrelated equations, for example, E_x can be written as

$$E_x^{n+1}\left(i + \frac{1}{2}, j, k\right) = E_x^n\left(i + \frac{1}{2}, j, k\right) + \frac{\Delta t}{\epsilon\left(i + \frac{1}{2}, j, k\right)} \times \left(\frac{H_z^{n+1/2}\left(i + \frac{1}{2}, j + \frac{1}{2}, k\right) - H_z^{n+1/2}\left(i + \frac{1}{2}, j - \frac{1}{2}, k\right)}{\Delta y} - \frac{H_y^{n+1/2}\left(i + \frac{1}{2}, j, k + \frac{1}{2}\right) - H_y^{n+1/2}\left(i + \frac{1}{2}, j, k - \frac{1}{2}\right)}{\Delta z} - J_x^n\left(i + \frac{1}{2}, j, k\right) \right) \quad (7.1)$$

It should be noted here that this set of equations is very well suited for parallel implementation and requires communication only between immediate neighbors. The third dimension for this simple case was done in the PE's memory. It was seen that the parallel version was over eleven times faster than the serial HP 735 code.

7.5 Discussion of Results

A baseline for comparison between serial and parallel code would be a single processor workstation built around a MasPar PE. Alternatively, it was decided that comparison would be made between the parallel MasPar code and the fastest, most optimized serial code on two locally available workstations. The speed-up for the 2D and 3D cases on the three platforms under different conditions are listed in Tables 7.2 and 7.3, respectively. It should be noted here that the serial HP 735 code was compiled with optimization options. The comparison of time taken to run was done by using the *time* command on serial versions and *gettimeofday* on the MP2.

It should also be noted that the random number generators provided with the MP2 system generated an unacceptably large number of repeated numbers and hence could not be used for Monte Carlo simulation. An alternative third party random number generator developed by Aluru et al. [89] showed proper behavior and was used instead.

The parallel MasPar program did not incorporate a number of optimizations which were available to the serial programs. Attempts to use SOR (successive over relaxation) led to instability and non-convergence of Poisson's equation. Further, many MasPar variables had to be declared double precision since declaring them single precision led to unacceptably large errors produced by numerical calculations. These two main differences also contributed in slowing down the parallel programs. Once SOR was disabled and double precision implemented in certain critical sections of the code, accurate and correct solutions were obtained on the MP2. However communication was not a major

Table 7.3: Results for the 3D simulation

Platform	Size	Time/ iteration in seconds	Speed-up Sun/HP	Normalized Speed-up Sun/HP	Efficiency % Sun/HP
Sun Sparc 10	32x32x32	122.92	–	–	–
HP 735	32x32x32	41.26	–	–	–
MasPar MP2 in memory (1024 PEs)	32x32x32	10.61	11.59/ 3.89	625.28/ 448.98	61/44
MasPar MP2 across PEs (4096 PEs)	32x32x32	3.065	40.1/ 13.46	2163.4/ 1553.6	53/38
MasPar MP2 across PEs (frequent data output to <i>loram</i>) (4096 PEs)	32x32x32	3.849	31.96/ 10.72	1724.2/ 1237.3	42/31
MasPar MP2 across PEs (graphical display) (4096 PEs)	32x32x32	3.1053	39.58/ 13.29	2135.3/ 1533.9	52/37
MasPar MP2 across PEs (graphical display) (16384 PEs)	64x64x64	7.4116	NA	NA	NA

bottleneck as was initially feared since the problem was more compute bound than communication bound and the communication was less than 8% in all cases.

The argument that similar portions in serial code be also made double precision and that SOR be disabled was discarded; it was concluded that the above two problems were actually a drawback of the MP2 and hence penalizing the serial machines for their better behavior was not correct. It should be further noted here that the 3D MP2 code did not output data to disk at the same frequency as the serial program. Versions of MP2 code writing the same amount of data to a fast *ioram* or, alternatively to graphical displays, did not exhibit significant deterioration in performance as can be seen in Table 7.3.

The results in Tables 7.2 and 7.3 show that speed-up of up to 13 times over the HP 735 and of over 40 times of a Sparc 10 can be achieved by properly designing the 3D parallel computation. It would also be interesting to compare the performance of serial and parallel machines equipped with similar processors. It is encouraging to note that even though the MFLOPS rating of the MP2 PE was much less than the Sparc 10 and considerably less than the HP 735's, the inherent parallelism of the application provides for tremendous speed-up over the serial case. The last row of Table 7.3 also shows the scalability of the problem. It can be seen that a plasma processing chamber eight times the size can be simulated with a slow-down of little over a factor of two. It should be noted here that a problem of this size could not be simulated in either of the serial machines due to unavailability of sufficient local memory.

For a workload W , the speed-up $S_n(W)$ with n processors to one processor is defined as

$$S_n(W) = \frac{T_1(W)}{T_n(W)} \quad (7.2)$$

where $T_i(W)$ is the time required to complete W amount of work on i processors. The speed-up presented here is the so-called *absolute speed-up* and is defined as the ratio of time required on a workstation (HP 735 or SUN Sparc) to the time required on a 1K or 4K MasPar. Because the processing power of a single processor of the MasPar is much lower than the HP or SUN workstation, the absolute speed-up is not a clear measure of the performance achieved by the MasPar parallel system. It is not possible to run the program on a single processor of the MasPar because of the memory limitation, and thus the serial program was run on a workstation. For the purposes here, the *normalized speed-up* is defined to be $S_n(W) * (\text{speed of workstation} / \text{speed of a MasPar PE})$. The workload W in this application is mostly floating point computation, so the manufacturers' MFLOPS ratings are used as the speed of interest.

The physical meaning of the normalized speed-up is the speed-up the parallel machine can achieve if its processors have the same processing power as the workstation. In an ideal case of linear speed-up, the normalized speed-up should equal the number of processors used (to be called ideal speed-up). In real applications, it is impossible to avoid the communication overhead among processors. The efficiency of a parallel system then is defined as the ratio of the normalized speed-up to the ideal speed-up. In Tables 7.2 and 7.3, the following numbers are reported: speed-up of the MasPar program with respect to the Sun and HP programs (separated by '/' in this column and the next two columns), the normalized speed-up, and the efficiency (as a percentage, where 100% is ideal). The numbers are for various programs on different numbers of PEs. While

the efficiency numbers indicate there may be some room for additional optimization, they are sufficiently high to judge the MasPar implementation as successful in terms of performance improvement.

The results of the simulations were viewed graphically using *MPDDL* and *gnuplot*. While *MPDDL* was easy to use, it lacked flexibility in display options and was not very useful for the programmers involved in this project. Color plots were generated locally from workstation versions of the code using *NCAR*, *MATLAB* and *gnuplot*. The availability of a fast disk allowed the design of a work-around for the parallel version. This involved writing data to disk while simultaneously an independent program, running on the front-end used *gnuplot* to provide run-time graphics. Overall, this chapter demonstrated the success with which particle-type plasma simulations can be parallelized.

Chapter 8

Distributed 3D3V Simulation

8.1 Introduction

The previous chapter demonstrated the implementation of plasma 2D3V and 3D3V simulations on a massively parallel processor computer. Since very few computing facilities have access to such specialized and expensive machines, it was theorized that a similar version of the simulation could be implemented across a group of single-processor machines. Further, since many of the laboratory machines undergo extended periods of idleness, it was thought that one could increase the overall usage of machines.

This chapter details the implementation of a distributed simulation across a group of Sun and HP workstations. This chapter also introduces a set of software routines named Parallel Virtual Machine (PVM) which was used in the implementation of the distributed application. The results of this study are also tabulated.

8.2 Parallel Virtual Machine

PVM is a software system that permits a network of heterogenous Unix computers to be used as a single large parallel computer [90]. Thus, theoretically, large computational problems can be solved using the aggregate power of many computers. Under PVM, a user defined collection of serial,

parallel, and vector computers appear as one large distributed memory machine. PVM provides a user interface and routines to start up, control, synchronize and communicate with tasks. Applications can be written in either Fortran or C, and PVM uses message passing constructs to communicate information amongst cooperating applications to solve a large problem in parallel.

PVM provides routines for packing and sending messages between tasks that make up the distributed application. Messages can be tagged and can be either task to task or global broadcasts. Similarly, tasks can await the receipt of messages by the message tags in a blocking or non-blocking manner. Further, there exists the ability for a receiving task to time out in a blocking receive and thus help break some potential deadlocks.

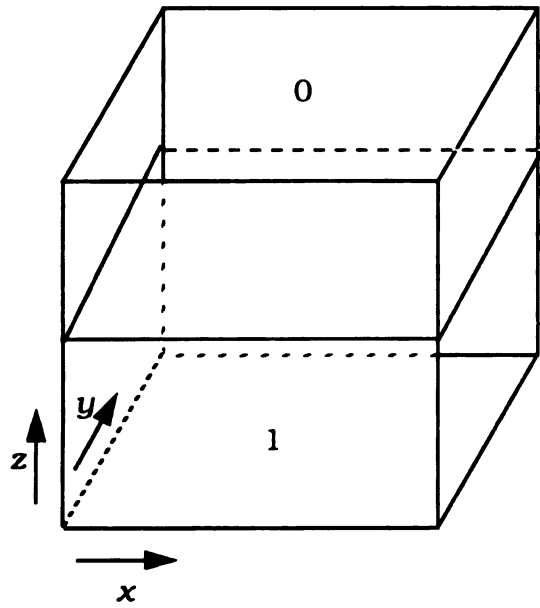
It should be cautioned here that PVM provides only very primitive communication and synchronization constructs and by itself does very little to aid the actual design of the distributed application. The actual design, control and fragmentation of the application has to be done independent of the message passing constructs used. Therefore, a well designed distributed application can be implemented using any one of the many distributed computing aids available of which PVM is one.

8.3 A Distributed Poisson's Equation Solver

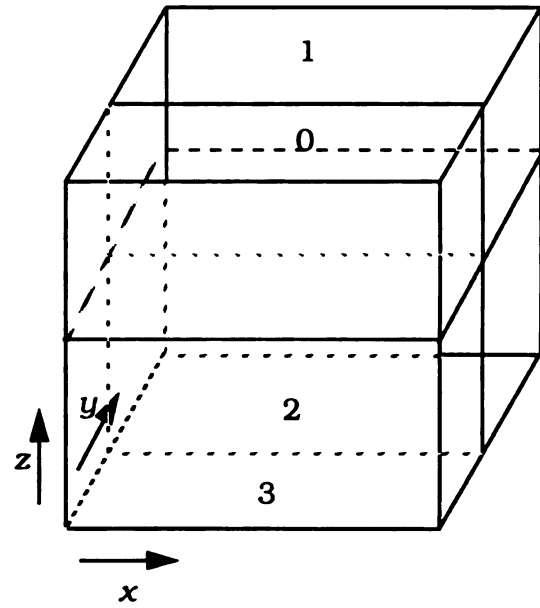
The pre-requisite to distributing a complete plasma simulation is the ability to design and implement a distributed solver for Poisson's equation. Designing a distributed Poisson's equation solver requires a very different perspective than the one used in the MPP case. In the implementation in the previous chapter, explicit synchronization between the processors was unnecessary since the

instructions were executed in lock-step across the many processors. However, distributing the application across many different laboratory machines presented an entirely different challenge. Even though it was decided to use a homogeneous set of machines with the same instruction speed and memory limitation for the application, the instantaneous load at any given time was unpredictable. Further, the initial and boundary conditions of the portion of the plasma being simulated can lead to different computational load on each of the machines. In addition to these challenges, the communication delays between the computers have also to be factored in. This leads to the conclusion that even under ideal conditions, explicit synchronization of the different portions of the distributed application is necessary. These requirements make the design of a distributed application more challenging than the MPP one.

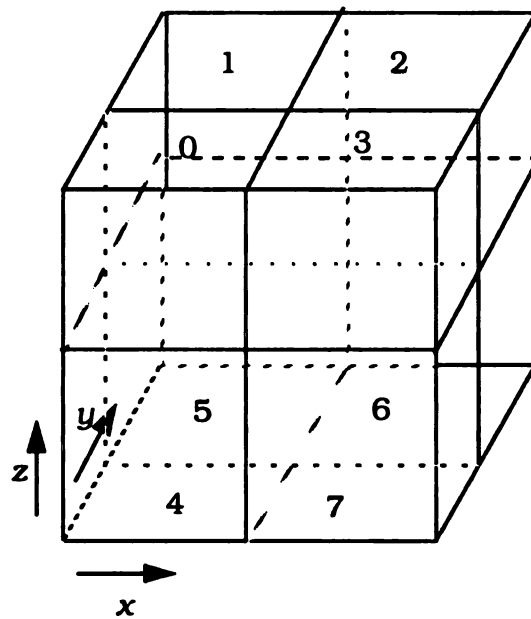
It was decided that the application would be divided across many computers in the order of powers of 2. For example, the serial $32 \times 32 \times 32$ grid structure could be broken up among two machines along the z axis leading to two $32 \times 32 \times 17$ grids on each of the machines. The scheme used by this approach to distribute applications is shown in Figure 8.1. The extra grid in the direction of the distribution is shared by the two computers in question as shown in Figure 8.2. The application was fragmented in this fashion since this maximized the number of faces which were held at some boundary condition thus reducing communications between machines. Figure 8.3 shows one of the many potential partitioning schemes which was discarded. In this version, the middle computer simulated a volume which had its upper and lower z planes shared with other machines. This would have lead to considerably more communication and possible instabilities.



2 Computers



4 Computers



8 Computers

Figure 8.1: Distribution of load across computers.

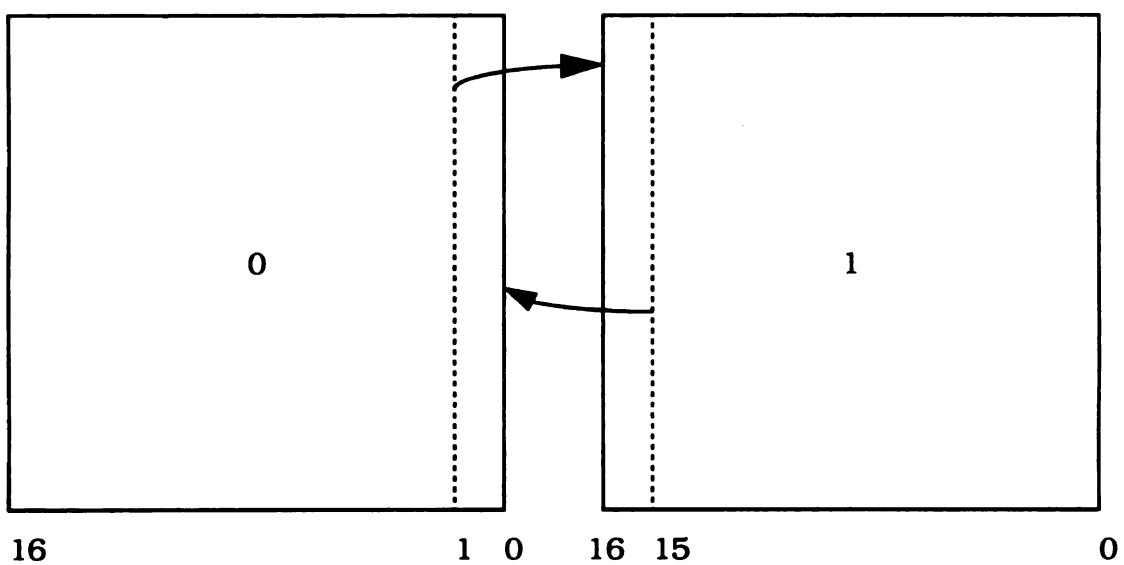


Figure 8.2: Shared planes across computers.

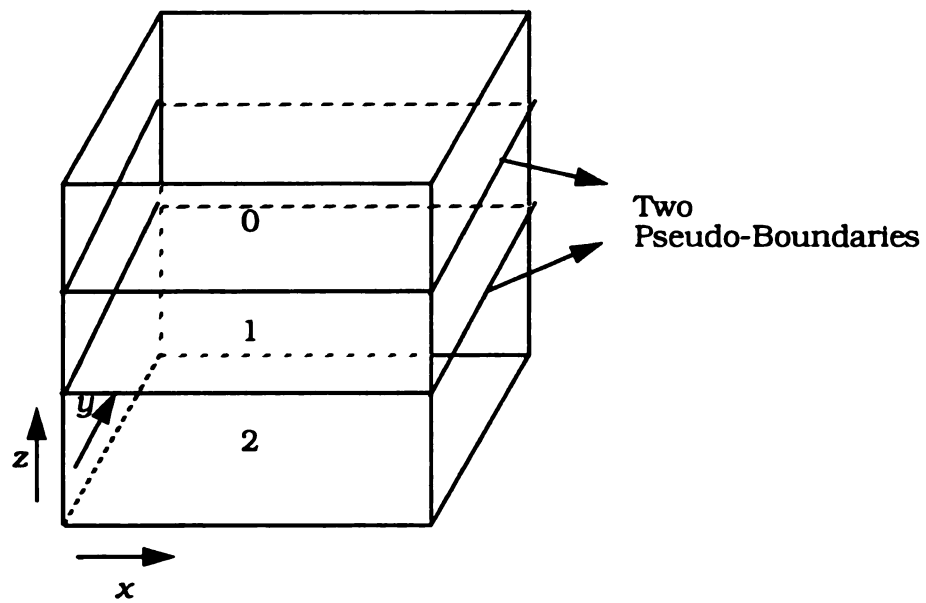


Figure 8.3: A discarded partitioning scheme.

It can be seen from Figure 8.1 that as the number of computers in the distribution increases, so does the amount of data being communicated. In the case of 8 computers, each processor simulates a volume that shares three faces with three different machines. The effect of this is to increase the overall communication to and from each individual machine. Further distributions were not implemented due to lack of many more similar machines and because it was considered that the sizeable communication overhead would offset any computational gains made.

Figure 8.2 shows the implementation of the Poisson's solver across two machines. The program iterates over only the internal points and considers all grids points on the extremities to be boundary points. However in this case, the last z plane for machine 1 and the first z plane for machine 0 are "pseudo-boundaries" and are shared between the two. Therefore, at the end of each Poisson solver iteration, machine 0 sends its 1st plane to machine 1 which receives this plane on its 16th plane and similarly, machine 1 sends its 15th plane to machine 0 which receives it on its 0th plane. This process is continued until both machines converge to a solution.

In order to implement scalability of this application, a master-slave type implementation was designed. The master program running on one of the computers is given all the input data and spawns the number of slaves required on as many different computers. The slaves do the actual task of solving the Poisson's equation. It is possible that one of the many slaves may come to a local convergence ahead of the other slaves. In order to ensure synchronization and prevent deadlock, all slaves communicate their convergence status to the master and continue with the task in hand. Once the master receives convergence

status from all the slaves, it broadcasts this to all slaves which exit from the solver routine as shown in the flow chart in Figure 8.4. The actual implementation of this solver was slightly more complicated since it required some modifications to prevent possible deadlock. The Poisson's equation solver was successfully implemented for the 2, 4 and 8 machine case providing a basis for building a distributed plasma application.

8.4 Distributed 3D3V Plasma Simulation Implementation

3D3V plasma simulations are extremely compute intensive as discussed in the previous chapter and most of the time is spent in movement of particles. It is theorized that by spreading this load across many machines, a definite improvement in speed can be achieved. However, in addition to the distributed Poisson's equation solver, the motion of particles across machines has to be accounted for and transferred. The loss of particles to the boundaries is simple and dealt with as before. However, the motion of particles across "pseudo-boundaries" involves transmitting considerable data involving the particle state across machines. In the case of two machines, the decision of the final location is very simple and any particle that crosses the pseudo-boundary is sent to the only other computer. However, in the four processor case, this becomes more complicated since each computer shares a face with two other computers and shares an edge with the remaining one. This adds to the already increased communication load due to the Poisson's solver. Figure 8.5 shows the flow chart of a distributed 3D3V plasma simulation. It can be seen that the Poisson's equation solver forms an essential component of the overall simulation. At the

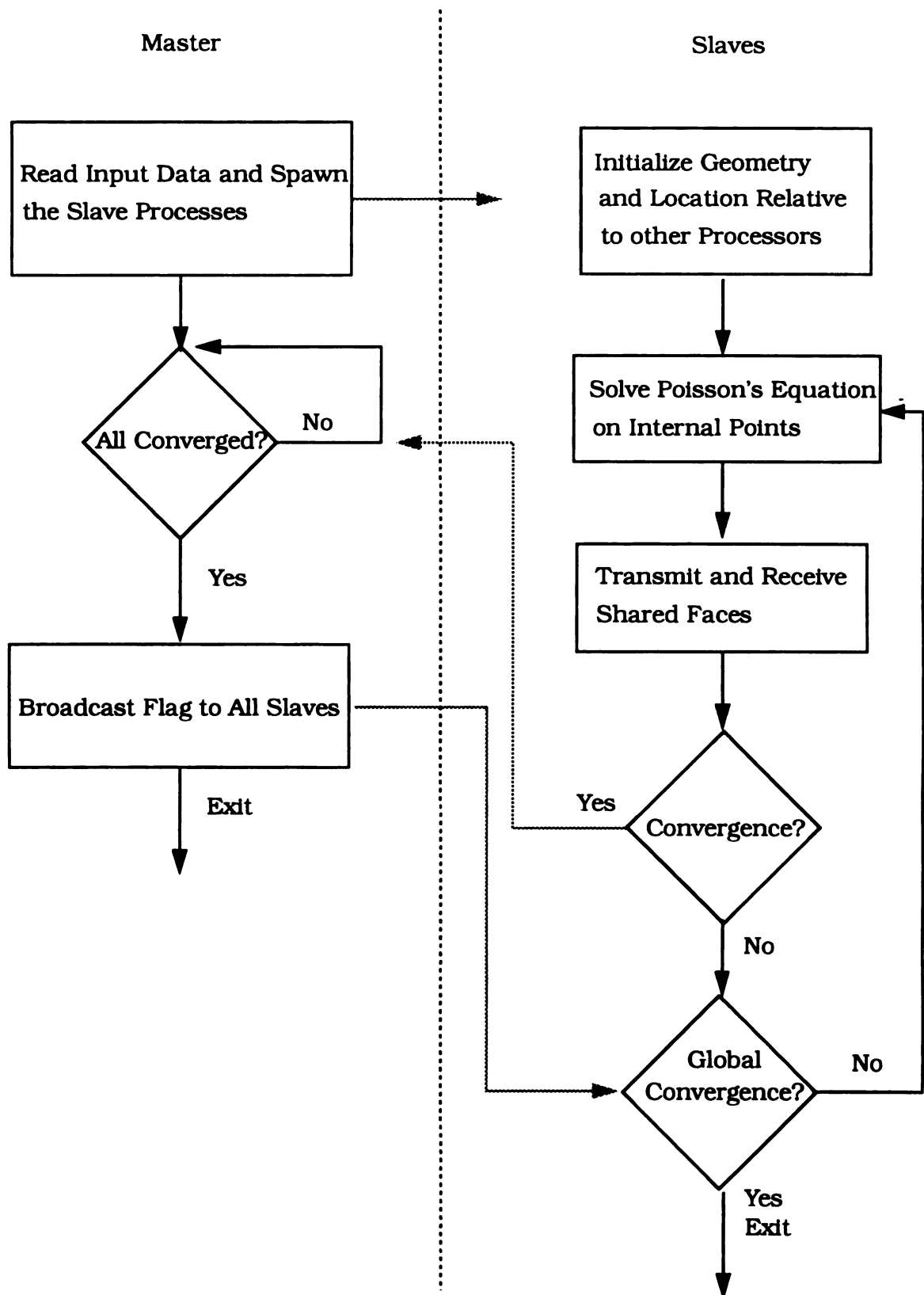


Figure 8.4: Flow chart of distributed Poisson's equation solver.

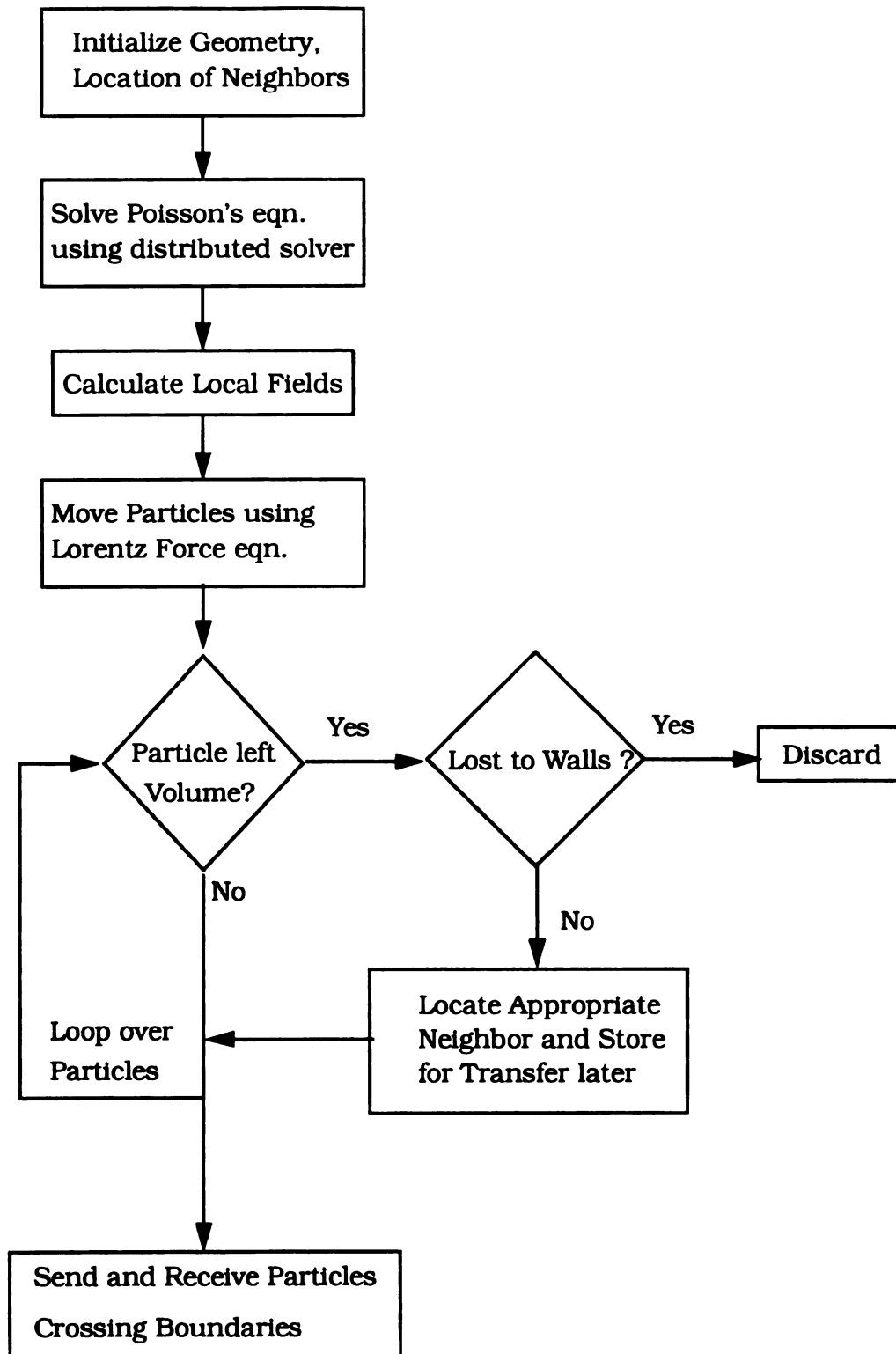


Figure 8.5: Flow chart of distributed 3D3V plasma simulation.

end of each iteration, the details of the particles crossing processor boundaries are transmitted to the corresponding computer. This process is continued till the whole system achieves a steady state. The eight processor implementation is much more complicated because in this case, the particles can move to one of the five potential destinations, three faces and two edges.

8.5 Discussion of Results

Time usage simulations for the distributed Poisson's equation solver was not done since it was expected to be slower than the one processor case due to its communication bound nature. In the case of the distributed plasma simulation it should be noted that the nature of the application provides considerable difficulty in obtaining exact timing details of the programs. In this case, one cannot use actual CPU usage as an indicator since during the time spent in waiting for communication, the UNIX kernel may decide to provide the CPU to some other unrelated process. This wait time that does not show up as CPU usage, should also be added to the actual CPU usage of the distributed applications since communication over-heads are an essential part of the timing analysis. The nature of this application makes it impossible to differentiate between communication induced delays and legitimate UNIX swap-out delays. However, one can get a measure of the communication to computation ratios by measuring the wall clock time used by the master program and comparing this with the CPU usage reported for the slaves. In the cases where communication tends to be the bottleneck, the slaves spend a large percentage of their time idling. Crowl [91] has discussed these problems in extensive detail. Further, he

described different methods to concisely present results of distributed computations.

3D3V simulations were conducted using $32 \times 32 \times 32$ grids for a single processor version, $32 \times 32 \times 17$ grids for the two processor version and $32 \times 17 \times 17$ grids for the four processor version. In order to minimize interruptions by unrelated processes, the application was run on Suns and HPs late at night in order to ensure as light a computational and network load as possible. The specifications of the Suns and HPs are given in Table 8.1. The results of the computations for two and four processors are given in Table 8.2. It can be seen that the two processor distributed version is nearly twice as fast for both the Suns and the HPs. However, contrary to what was hoped, the four processor version is much slower than the two processor version for the HPs and marginally slower for the Suns. This is due to the fact that while the processing speed of the machines have increased tremendously over the past few years, the communication backbone has remained virtually the same. In the case of the laboratory machines used in this research, the background network traffic is typically high. Further, it should be noted that these machines are connected in a serial fashion and therefore, communication between machines that are not physically one hop away tend to be more disadvantaged. All this is substantiated by the fact that actual CPU usage of each of the slave tasks was less than a quarter of the total wall clock time used leading to the conclusion that the processes are waiting for communication.

In view of the above results, it can be concluded that distribution of computation across several computers has a definite advantage, however, this can be negated due to the lack of sufficient communication bandwidth.

Table 8.1: Specification of the two systems

Platform	Clock Speed	MIPS Rating	MFLOPS Rating	SPECint92	SPECfp92
Sun Sparc 10	36 MHz	101.6	20.5	45.2	54
HP 715	80 MHz	-	-	83.5	120.9

Table 8.2: Results of distributed simulation

Platform	Secs/iteration (1 processor)	Secs/iteration (2 processors)	Secs/iteration (4 processors)
Sun Sparc 10	49.47	26.56	29.77
HP 715/75	29.07	14.53	24.23

Therefore, it was decided to conduct the same set of simulation runs on a set of systems connected by a high speed network.

The set of systems chosen for this purpose were a set of Sparc 10s connected on a high speed Asynchronous Transfer Mode (ATM) network available at the High Speed Network Processing laboratory in the Computer Science Department at Michigan State University. One further advantage that this set of machines had was that they were connected on a sub-net thus shielding them from the background communication traffic on the main network. Therefore, it was considered instructive to run the set of simulations on one, two and four sets of sub-netted machines using the regular communication channels. The result for this set of simulations is given in the first row of Table 8.3 and it can be clearly seen that there is a definite improvement in the 4 machine case over the two machine case. However, these results do not seem to justify the extra number of machines needed for the speed-up.

The same set of simulations were done using the high speed ATM network for communicating data between the processes. The results of this set of simulation runs is given in the second row of Table 8.3. However, the results were quite disappointing because the speedup provided by the high speed network was marginal for this particular problem. This could be due to the small size of the messages broadcast by the distributed application and hence this application does not take advantage of the high throughput capability of the ATM network. A comprehensive comparison for solving matrix multiplication and partial differential equations using a number of different Application Programming Interfaces (APIs) has been done by Lin et al [92]. In their work, they compared the distributed implementations of the two applications using Fore System's

Table 8.3: Results on a sub-netted system

Platform	Secs/iteration (1 processor)	Secs/iteration (2 processors)	Secs/iteration (4 processors)
Sun Sparc 10 (<i>low speed network</i>)	48.79	25.33	18.2
Sun Sparc 10 (<i>ATM network</i>)	–	24.38	17.38

ATM API, BSD socket programming interface, Sun's Remote Procedure calls and PVM. Their performance parameters included maximum achievable throughput and round trip timing and the results clearly show that for small packet sizes (< 2 KBytes), there was no significant difference in the performance of the four interfaces. However, a choice of larger packet sizes led to higher throughput and lower round trip timing for the ATM interface.

For the example case implemented using PVM, the maximum message size was less than 2 KBytes in all cases. Further, the size of the message decreases as the number of processors increases but the number of messages increases with an increase in the number of processors leading to more communication bottleneck. In view of these results, an eight processor distributed implementation was not considered. A comprehensive overview of the results can be seen in Figure 8.6. It is clearly shown that the communication overhead effects a faster system to a larger extent than a slower system. PVM also allows for tasks to set up direct TCP links between each other and this option has been used [92] to achieve better performance. This option could be enabled only on the HPs and the results are shown in Figure 8.6 under the legend "HP_DIR". It can be seen that there is a definite improvement over the default case which uses the PVM daemon as a communication interface. It can be concluded that this option can lead to higher performance in sub-netted systems. It is possible to reduce the communication overhead involved in solving Poisson's equation by reducing the number of messages by grouping them together to form one large packet. However, this will involve re-designing the algorithm completely to take advantage of throughput.

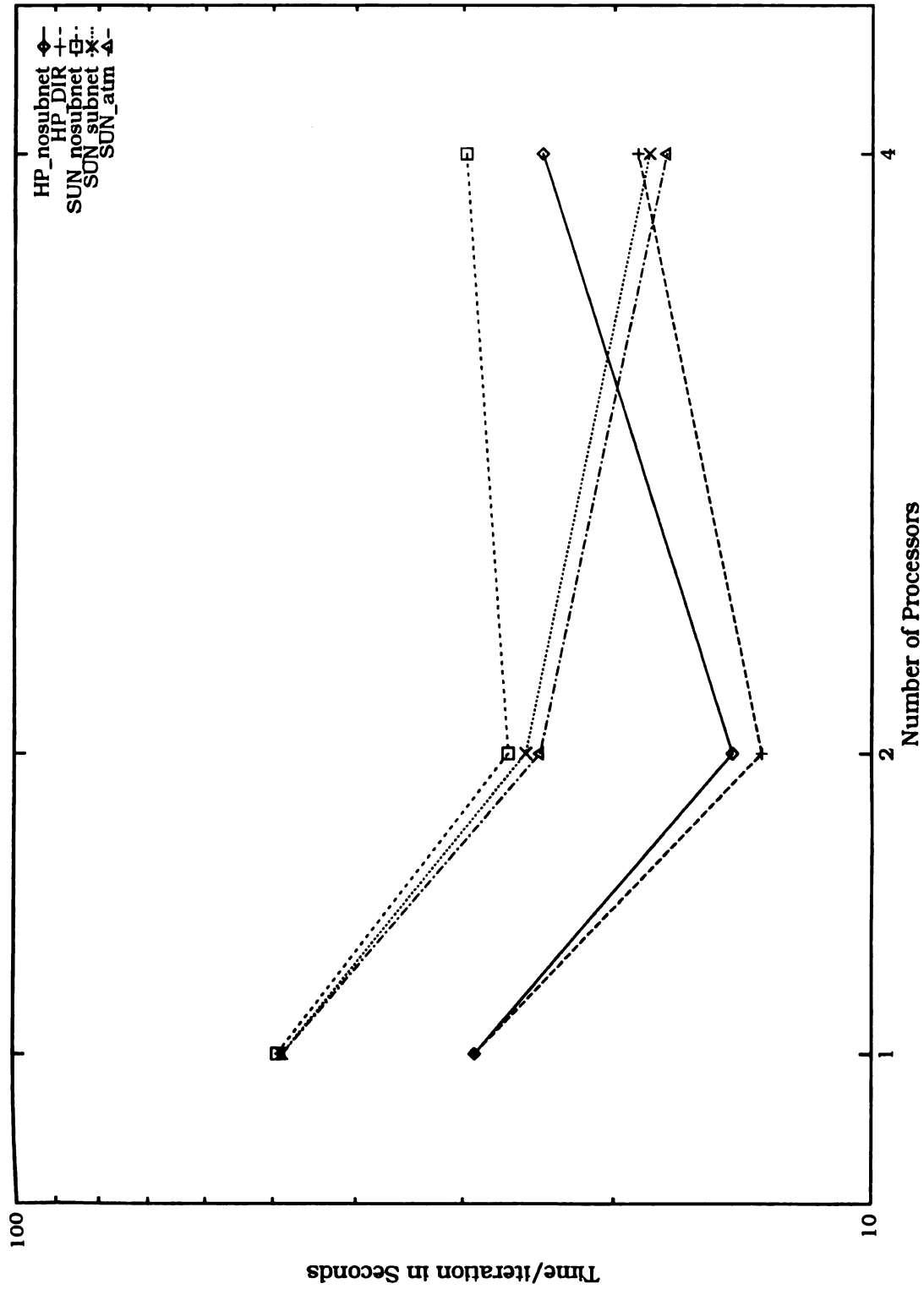


Figure 8.6: Comprehensive view of results.

Chapter 9

Conclusion

9.1 Summary of Research

Simulation techniques for the modeling of microwave plasma sources have been investigated and improved in this dissertation. The difficulties associated with two and three dimensional plasma source simulations which self-consistently solve the plasma behavior and electromagnetic/electrostatic fields were identified and solutions to the difficulties tested. The difficulties investigated included simulation instabilities in cylindrical coordinates, large computer memory requirements and long computer simulation time. Two simulation programs (2D3V and 3D3V particle models) were developed to investigate these difficulties and their solution. The simulation programs were applied to two different simulation examples including a compact ECR ion/free radical source and the downstream plasma processing region of a ECR plasma etching machine.

A self-consistent 3D3V plasma-EM numerical model has been developed to simulate discharges inside the microwave plasma source. The software couples an electromagnetic field model and a particle plasma model to simulate low pressure ECR plasmas. The primary particle collision mechanisms were included to realistically simulate plasma dynamics. The plasma source investigated was a 3cm inner diameter compact ion/free radical source used in

MBE machines. The microwave power absorption pattern was simulated and demonstrated to be at the expected ECR zones. A staggered grid system was developed to improve numerical stability in cylindrical structures. The model was applied to argon plasma simulation, although other discharge types can also be simulated. The input parameters to the simulation included operating pressure and magnetic field pattern. Characteristics of the plasma including plasma temperature, charge density, plasma potential, radial electric field distribution near the walls and power absorption patterns were also investigated. Further, simulations were modified to take advantage of parallelism between the plasma and EM programs to produce considerable speedup.

A 2D3V simulation of the downstream region of a plasma processing machine was conducted assuming ϕ symmetry. The plasma source was assumed to provide a constant flux of particles to the downstream region. The plasma potential and the charge densities across the simulation volume were resolved. A non-uniform particle weighting scheme was implemented to aid stability in the grid points closer to the origin and was shown to be successful.

Both the 3D3V compact ion source and the 2D3V downstream plasma processing region simulations were limited in their ability to model only small volumes at reasonably low densities because of the memory limitation of the single CPU workstation used. In order to extend the 3D3V and 2D3V simulations to large plasma machines operating at higher densities, parallel computing and distributed computing techniques were investigated.

The movements of particles in particle type plasma simulations are essentially independent of each other once the governing forces have been self-consistently determined. Therefore, implementations of 2D3V and 3D3V plasma

simulations on a MasPar MP2 massively parallel processor were performed. Speed-up of over 13 times the fastest locally available serial machine was achieved, demonstrating the possibility of parallel computing simulations. Further, simulation of problems sizes beyond the computing and storage capability of desktop machines was also demonstrated. Different methods of simulating the third dimension in order to achieve maximum speed-up were also demonstrated.

The concept of parallel simulations was modified to spread the computation load across a set of networked desktop machines. Since very few institutions can afford access to a MPP, it was postulated that considerable speed-up could be gained by dividing an application across a number of similar workstations. It was also thought that the speed-up gains would offset the loss due to increased interprocessor communications. A distributed simulation using PVM was designed to run across 2, 4 and 8 workstations. The two processor implementation was shown to provide considerable speed-up over a single desktop case. However, communication bottlenecks reduced the gain considerably for the four processor implementation.

Overall, the objective of performing 2D3V and 3D3V simulations of microwave plasma ECR sources and machines was achieved. Contributions were made in improving the stability of cylindrical co-ordinate simulations, and handling computer memory and CPU time requirements by using parallel and distributed computing.

9.2 Directions for Future Research

The complete plasma-Maxwell equations show considerable promise in simulating microwave processing cavities used in the laboratory. An example simulation was applied to a relatively low density argon plasma. Computer speed and memory limitations prevent the application of this model to higher densities and larger cavities on a single CPU workstation. It is expected that the availability of faster and larger computer resources in the near future will allow the simulations of larger plasma machines.

Further, the simulation includes only the most basic collisions, gas types and two charged particles. This should be extended to include more complex collisions, many ions and a mixture of gas types. This will enable prediction of various plasma component densities and fluxes.

The downstream simulation developed in this research was also applied to a small example system at lower plasma densities. Here again, the lack of computer resources hampered the simulation of a larger system. This system is well suited to be implemented on an Massively Parallel Processor. Further, implementations on an MPP will allow simulations beyond the range and capacity of conventional desktop machines.

The 3D3V MPP simulations successfully demonstrated in this research were for regular rectangular structures. Extensions of this idea to simulate complex structures and geometries hold considerable promise. The PVM version of parallel simulations have been shown to aid parallelization of plasma-EM simulations. This concept should be extended to distribute both components of the simulation across multiple workstations. Further research needs to be done

regarding communication bottlenecks and methods to reduce these in distributed plasma implementations.

Lastly, the utility of these simulation programs can be harnessed in the future for the design and optimization of microwave plasma machines and processes.

List of References

List of References

- [1] J. Asmussen, J. Vac. Sci. Technol. A, **7**, 883 (1989).
- [2] D. M. Manos and D. L. Flamm, "*Plasma Etching, An Introduction*," Academic Press, 1989.
- [3] J. Hopwood, M. Dahimene, D. K. Reinhard, and J. Asmussen, J. Vac. Sci. Technol. B **6**, 268, (1988).
- [4] G. T. Salbert, "Anodic Growth And Cathodic Removal of Silicon dioxide layers utilizing an Electron Cyclotron Resonant Microwave Plasma Disk Reactor," Ph.D. Dissertation, Michigan State University, 1992.
- [5] Gopinath, V., G. T. Salbert, T. A. Grotjohn, and D. K. Reinhard, "ECR Sputter Removal of SiO₂ on Silicon Wafers," J. Vac. Sci. Technol. B **11**(6) Nov/Dec 1993.
- [6] J. A. Bittencourt, "*Fundamentals of Plasma Physics*," Pergamon Press, 1988.
- [7] C. K. Birdsall and A. B. Langdon, "*Plasma Physics via Computer Simulation*," McGraw Hill, N. York, 1981.
- [8] T. A. Grotjohn, "Numerical modeling of a compact ECR ion source," Rev. Sci. Instrum., **63** (4), Apr. 1992, pp. 2535-2537.
- [9] V. Vahedi., R. A. Stewart, and M. A. Lieberman, "Analytic Model of the Angular Distribution in a Collisional Sheath," 39th National AVS Symposium and Topical Conference, Chicago, Nov. 1992.
- [10] H. X. Vu, Brackbill J. U., "CELEST1D: an implicit fully kinetic model for low-frequency, electromagnetic plasma simulation," Comp. Phys. Communication, Vol. **69**, pp. 253-76, Mar-Apr 1992.
- [11] J. U. Brackbill, and D. W. Forslund, "An Implicit Method for Electromagnetic Plasma Simulation in Two Dimensions," J. Comp. Phys., **46**, (1982), pp. 271-308.

- [12] Trombley H. W., Terry F. L. Jr., Elta M. E., "A Self-Consistent Particle Model For the Simulation of RF Glow Discharges," IEEE Transactions on Plasma Science, Vol. 19, No 2, Oct. 1991, pp. 158-162.
- [13] Vender D., Boswell R. W., "Numerical Modelling of Low-Pressure RF Plasmas," IEEE Transactions on Plasma Science, Vol. 18, No 4, Aug. 1990, pp. 725-732.
- [14] Sprott J. C. and Edmonds P. H., "Computer Calculations of Electron Cyclotron Heating in a Nonuniform Magnetic Field," The Physics of Fluids, Dec. 1971, pp. 2703-2707.
- [15] Sadeghi N., Nakano T., Trevor D. J., and Gottscho R. A., "Ion Transport in an Electron Cyclotron Plasma," J. Appl. Phys., 1991.
- [16] M. Hussein and Emmert G. A., "Numerical Simulation of ECR microwave plasma stream," International Conference on Plasma Science, pp. 147, 1989.
- [17] Shagirov E. A., "Computation of three-dimensional magnetic fields and particle trajectories in the presence of ferromagnets," Computational Mathematics and Modeling, Vol. 2 pp. 458-61, Oct-Dec. 1991.
- [18] S. Brandon, D. J. Larson, N. Madsen, D. E. Nielsen Jr., and P. Weidhaas, "3-D Electromagnetic Plasma Simulation Using Non-Orthogonal Unstructured Grids," presented at the IEEE conference on Plasmas, 1993.
- [19] W. Tan and T. A. Grotjohn, "Modeling The Electromagnetic Excitation of Microwave Resonant Cavities Used For Plasma Discharge Generation," Submitted to IEEE Transactions on Plasma Science.
- [20] C. S. Lin, A. L. Thring, Koga J., and E. J. Seiler, "A parallel particle-in-cell model for the Massively Parallel Processor," Journal of Parallel and Distributed Computing, Vol. 8 pp. 196-9, Feb. 1990.
- [21] R. Lohner and J. Ambrosiano, "A Vectorized Particle Tracer for Unstructured Grids," J. Comp. Phys., 91, pp. 22-31 (1990).
- [22] S. Matsushita, Narusawa M., Kurita G. et. al., "Parallelization of nonlinear MHD plasma simulation," Transactions of Information Processing Society of Japan, Vol. 33, p 360-8.

- [23] L. J. Mahoney, "The Design and Testing of a Compact Electron Cyclotron Resonant Microwave-Cavity Ion Source," M. S. Thesis, Michigan State University, 1989.
- [24] A. K. Srivastava, M. Dahimene, T. Grotjohn, and J. Asmussen, "Experimental Characterization of a compact ECR ion source," *Rev. Sci. Instrum.* 63(4), pp. 2556-2558, Apr. 1992.
- [25] Gopinath V., and T. A. Grotjohn, "ECR Ion/Free-Radical Plasma Source Simulation in Two and Three Dimensions," presented at the 1994 IEEE International Conference on Plasma Science, Santa-Fe, June 1994.
- [26] V. Gopinath, T. A. Grotjohn, D. Rover, and Y-K Chu, "Parallelization and Performance of Three-Dimensional Plasma Simulation," submitted to *Frontiers '95*.
- [27] V. Gopinath, T. A. Grotjohn, D. Rover, and Y-K Chu, "Three-Dimensional Plasma Simulations using a Massively Parallel Processor computer," to appear in the Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing.
- [28] D. W. Hess, "Plasma Material Interactions," *J. Vac. Sci. Technol. A* (8), pp. 1677-1685, May/June 1990.
- [29] S. Ramo, J. R. Whinnery, and T. VanDuzer, "*Fields and Waves in Communications and Electromagnetics*," John Wiley and Sons, N. Y. 1984.
- [30] T. Tajima, "*Computational Plasma Physics: with applications to fusion and astrophysics*," Addison-Wesley, 1989
- [31] Hammersley J. M., "*Monte Carlo Methods*," Methuen, London, 1964.
- [32] C. Jacoboni and P. Lugli, "*The Monte Carlo Method for Semiconductor Simulations*," Springer-Verlag, New York, 1989.
- [33] Thompson, B.E., Swain H. H., Fisher D. A., "Monte Carlo simulation of ion transport through rf glow-discharge sheaths," *J. Appl. Phys.*, 63 (7), Apr. 1988.
- [34] Govindaraju G. R., "Monte Carlo Simulation of Electron Swarms in Nitrogen in Uniform $E \times B$ fields," *IEEE Transactions on Plasma Science*, Vol. 18, No 5, Oct. 1990, pp. 819-825.

- [35] J. Ignacio, Ulacia F. and James P. McVittie, "A two-dimensional computer simulation for dry etching using Monte Carlo techniques," J. Appl. Phys. 65(4), Feb. 1989.
- [36] J.P. Biersack and W. Eckstein, "Sputtering Studies with the Monte Carlo Program TRIM.SP," Appl. Phys. A **34**, 73-94 (1984).
- [37] D. B. Graves, "Fluid Model Simulations of a 13.56 MHz rf discharge: time and space dependence rates of electron impact excitation," J. Appl. Phys. 62(1), pp. 88-94, July 1987.
- [38] M. S. Barnes, T. J. Colter, and M. E. Elta, "Large-signal time-domain modeling of low-pressure rf glow discharges," J. Appl. Phys. 61 (1), Jan. 1987.
- [39] Bandyopadhyay S. et. al., "A Rigorous Technique to Couple Monte Carlo and Drift-Diffusion Models for Computationally Efficient Device Simulation," IEEE Transactions on Electron Devices, Vol. ED-34, No 2, Feb 1987.
- [40] J. K. Lee, C. K. Birdsall, "Particle-fluid hybrid simulations for weak beam-plasma interactions," Comp. Phys. Comm., 64 (1991) pp. 214-220.
- [41] W. Tan, Ph. D. Dissertation, Michigan State University, 1994.
- [42] M. Surendra, D. R. Graves, and L. S. Plano, "Self-consistent dc glow-discharge simulations applied to diamond film deposition reactors," J. Appl. Phys. 71 (10), May 1992.
- [43] Birdsall D. K., and D. Fuss, "Cloud-in-clouds, clouds-in-cells physics for many body simulations", J. Comput. Phys., 3, pp. 494-511, Apr. 1969.
- [44] A. B. Langdon, "On Enforcing Gauss's law in Electromagnetic Particle-in-cell codes," Computer Physics Communications, 70 (1992) 447-450.
- [45] B. Marder, "A Method for Incorporating Gauss's Law into Electromagnetic PIC Codes," J. Comp. Phys., **68**, pp. 48-55 (1987).
- [46] D. E. Nielsen, Jr. and A. D. Drobot, "An Analysis and Optimization of the Pseudo-current Method," J. Comp. Phys., **89**, pp. 31-40 (1990).
- [47] Hese J. V., and Zutter D. D., "Modeling of Discontinuities in General Coaxial Waveguide Structures by the FDTD Method," IEEE Transactions on Microwave Theory and Techniques, Vol. 40, No 3, Mar. 1992

- [48] A. Matsumoto, S. Kawata, "TRIPIC: Triangular-Mesh Particle-in-Cell Code," J. Comp. Phys., 87, pp. 488-493 (1990).
- [49] E. Sonnendruker, J. Ambrosiano, S. Brandon, "A Finite Element Formulation of the Darwin Electromagnetic PIC Model for Use on Unstructured Meshes," presented at the IEEE conference on Plasmas, 1993.
- [50] Schottky, W. Phys. Z., 25, 635 (1924) as cited by [23].
- [51] J. Asmussen, "EE850 Class notes".
- [52] G. A. Geist, V. S. Sunderam, "Network-based Concurrent computing on the PVM system," Concurrency: Practice and Experience, Vol: 4, pp. 293-311.
- [53] S. Y. Liem, Brown D. and Clarke J. H. R., "Molecular Dynamics on Distributed memory machines," Comp. Phys. Comm., 67 (1991) pp. 261-267.
- [54] F. Muller-Plathe, "Parallelizing a molecular dynamics algorithm on a multi-processor work station," Comp. Phys. Comm., 61 (1990) pp. 285-293.
- [55] Beguelin, A. et. al., "Graphical Development tools for network-based concurrent supercomputing," Proceedings, Supercomputing '91, pp. 435-44.
- [56] R. E. Ewing, R. C. Sharpley, D. Mitchum, P. O'Leary and J. S. Sochaki, "Distributed Computation of Wave Propagation Using PVM," IEEE Parallel and Distributed Technology, Systems and Applications, Spring 1994, pp. 26-31.
- [57] P. C. Liewer and V. K. Decyk, "A general concurrent algorithm for plasma particle-in-cell simulation codes," J. Comp. Phys., Vol. 85, pp. 302-22, Dec 1989.
- [58] A. F. Kuckes, "Resonant Absorption of Electromagnetic Waves in a Non-Uniformly Magnetized Plasma," Plasma Physics, Vol. 10, pp. 367-380, 1968.
- [59] P. K. Trost and J. L. Shohet, "Electron behavior during electron cyclotron resonance heating in a stellarator," Phys. Fluids B 1 (6), June 1989.

- [60] J. C. Sprott, "Electron Cyclotron Heating in Toroidal Octupoles," *Phys. Fluids*, Vol. 14, **8**, Aug 1971.
- [61] M. A. Lieberman and A. J. Lichtenberg, "Theory of Electron Cyclotron Resonance Heating-II. Long Time and Stochastic Effects," *Plasma Physics*, Vol. 15, pp. 125 - 150, 1973.
- [62] M. Hussein and G. A. Emmert, "Collisional effects on plasma flow along the divergent magnetic field of an ECR plasma stream source," *J. Vac. Sci. Technol.*, Vol. 8A, no **3**, pp. 2913-2918, 1990.
- [63] S. A. Self and H. N. Ewald, "Static theory of a discharge column at intermediate pressures," *Phys. Fluids*, Vol. 9, no 12, pp. 2486-2492, 1966.
- [64] Porteus R. K., and Graves D. B., "Modeling and Simulation of Magnetically confined Low-Pressure Plasmas in Two Dimensions," *IEEE Transactions on Plasma Science*, Vol. 19, No 2, Apr. 1991, pp. 204-213.
- [65] M. Hussein and G. A. Emmert, "Modeling of ion energy spectrum in electron cyclotron resonance plasmas," *IEEE Conf. on Plasma Science*, 1991.
- [66] Weng. Y. and Kushner Y. J., "Simulation of remote plasma activated processing using electron cyclotron excited discharges," *IEEE Conf. on Plasma Science*, 1990.
- [67] Graves D. B., Wu H. and Porteus R. K., "Modeling and Simulation of High Density Plasmas," *Jpn. J. Appl. Phys.* Vol. 32 (1993) pp. 2999-3006.
- [68] Y. Yasaka, A. Fukuyama, A. Hatta and R. Itatani, "Two-dimensional modeling of electron cyclotron resonance plasma production," *J. Appl. Phys.* **72** (7) Oct. 1992, pp. 2652-2658.
- [69] S. Salimian, C. B. Cooper, III, and A. Ellingboe, *Appl. Phys. Lett.*, 56, 1311, (1990).
- [70] J. L. Vossen, "*Physics of Thin Films: Contemporary Preparative Techniques*," edited by M. H. Francombe and J. L. Vossen (Academic Press, Boston, 1989), pp. 201-240.
- [71] P. Raymaud and C. Pomot, *J. Vac. Sci. Technol. B*. Vol. 11, pp. 699 (1993)

- [72] P. Mak, G. King, T. A. Grotjohn, and J. Asmussen, *J. Vac. Sci. Technol. A* 10(4), 1281, (1992).
- [73] G. King, F. C. Sze, P. Mak, T. A. Grotjohn, and J. Asmussen, *J. Vac. Sci. Technol. A* 10(4), 1265, (1992).
- [74] Grotjohn T. A., V. Gopinath, A. K. Srivastava, and J. Asmussen, "Modeling and Characterization of Hydrogen and Helium Discharges in a Compact ECR Ion/Free Radical Source," Fifth International Conference on Ion Sources, Beijing, August 1993.
- [75] A. K. Srivastava, J. Asmussen, "Measurements of the Impressed Electric Field inside a Coaxial Electron Cyclotron Resonance Source," to be published.
- [76] F-C. Sze, Ph.D. Dissertation, Michigan State University, 1993.
- [77] Y. P. Raizer, "*Gas Discharge Physics*," Springer-Verlag, 1991.
- [78] R. K. Janev, W. D. Langer, K. Evans Jr., and D. E. Post, Jr., "*Elementary Processes in Hydrogen-Helium Plasmas, Cross Sections and Rate Coefficients*," Springer-Verlag, Berlin, 1987.
- [79] B. E. Cherrinton, "*Gaseous Electronics and Gas Lasers*," Pergamon press, Oxford, 1979.
- [80] B. Chapman, "*Glow Discharge Processes: Sputtering and Plasma Etching*," John Wiley and Sons, New York, 1970.
- [81] C.-H. Wu, X. Wu, J.-H. Tsai, F. F. Young and C. Li, "Two-Dimensional Kinetic and Fluid Models for Parallel-Plate RF Glow Discharges," presented at the 1994 IEEE International Conference on Plasma Science, Santa-Fe, June 1994.
- [82] Tim Busse, MasPar Computer Corporation, Personal Communications.
- [83] MPL reference manual, Software version 3.2, Rev. A4, May 1993, MasPar Computer Corporation.
- [84] MPPE user guide, Software version 3.0, Rev. A8, July 1992, MasPar Computer Corporation.
- [85] Online MasPar commands reference man pages.

- [86] Hewlett-Packard product catalogue, <HREF =<http://www.wsg.hp.com/wsg/Global/Products/S700Workstations/HighEnd/Model735.html>>
- [87] Sun product catalogue, Rev. 5/11/93, pp. 12.
- [88] MasPar Data Parallel Programming Guide and miscellaneous handouts.
- [89] S. Aluru, G. M. Prabhu, and J. Gustafson, "A Random Number Generator for Parallel Computers," *Parallel Computing*, Vol. 18, Aug 1992.
- [90] PVM 3 User's Guide and Reference Manual, A Geist et. al., May 1994.
- [91] Lawrence A. Crowl, "How to Measure, Present, and Compare Parallel Performance," *IEEE Parallel and Distributed Technology, Systems and Applications*, Spring 1994, pp. 9-25.
- [92] M. Lin, J. Hsieh, D. H. C. Du, J. P. Thomas, and J. A. MacDonald, "Distributed Network Computing over Local ATM Networks," to appear in *IEEE Journal on Selected Areas in Communications Special Issue of ATM LANs: Implementations and Experiences with an Emerging Technology (Early '95)*.
- [93] Srivastava A., Private communications, Ion density vs. Position measurements for [24], not yet published.

MICHIGAN STATE UNIV. LIBRARIES



31293010289365