Chapter 2

Particle-in-cell Techniques

The Particle-in-cell (PIC) technique is a well-established method of modelling plasmas, and has been used for approximately the last 15 years to simulate low pressure rf reactors. Several books (especially Birdsall and Langdon (1985) and Tajima (1989)) detail the methods used in PIC and describe the advantages and limitations of the technique, therefore this thesis will only present a brief outline of the methods used, concentrating on modifications required by the spherical geometry and the external circuit. Birdsall and the computational plasma physics group at University of California, Berkeley pioneered the use of PIC techniques for low-pressure, bounded plasma simulations and so the methods detailed in Birdsall's book are followed most closely.

The system modelled by the simulation consists of two concentric spherical electrodes, separated by 20cm gap in which the plasma is formed, and an external circuit. This is shown schematically in Figure 2.1. The inner electrode is connected to an rf source

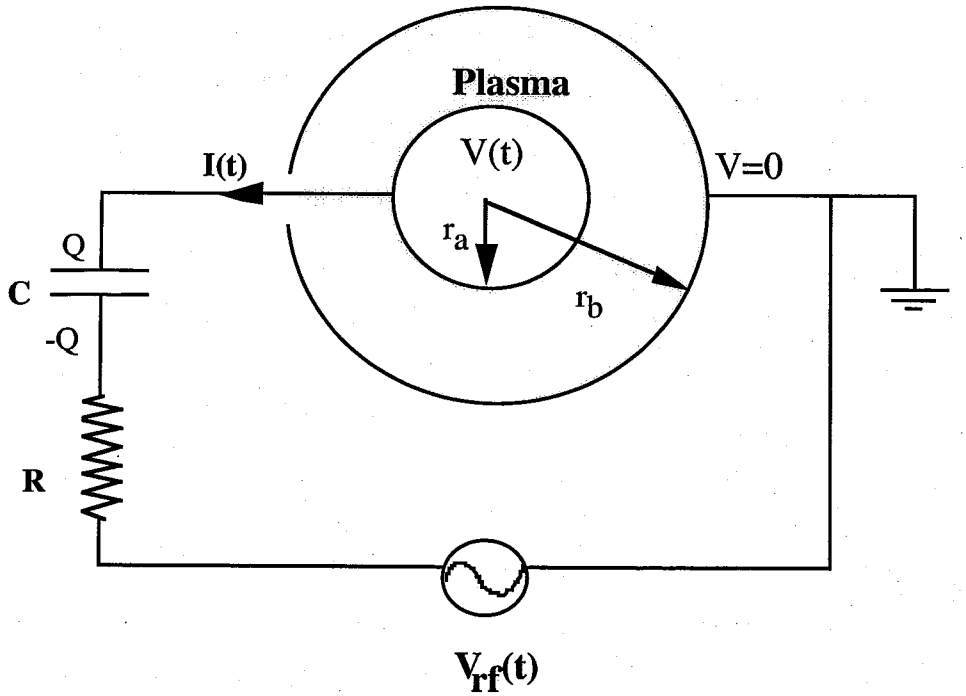

Figure 2.1 Schematic of plasma and external circuit, where r_a and r_b are the radii of the inner and outer electrodes respectively, $V(t)$ is the voltage on the inner, powered electrode and $V_{rf}(t)$ is the rf source voltage.

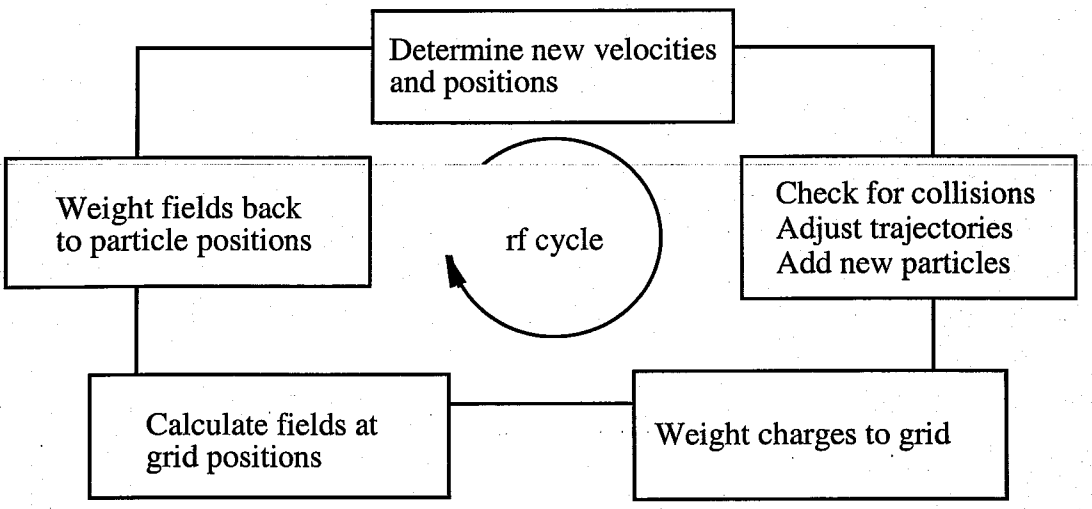

Figure 2.2 A schematic showing the algorithm for one time-step of the simulation.

Even using superparticles it is too time-consuming to calculate the inter-particular forces directly, since the number of calculations vary as the square of the number of simulation particles, which means 50,000 particles would involve 2.5×10^9 calculations each time the particles are moved! Hence, in the early 1960s, as reviewed in Birdsall (1991), a mathematical mesh or grid across the length of the plasma was introduced for this type of numerical modelling creating the first true "particle-in-cell" simulations. In PIC the charge densities are determined at the grid points by interpolating charge from the particle positions, and Poisson's equation is solved at the grid using finite difference methods. Electric fields on the grid calculated from the potentials can then be interpolated back to the particle positions, and Newton's equations used to move the particles. A diagrammatic representation of one time-step of the simulation is shown in Figure 2.2.

It should be noted here that introducing a grid has two major purposes – not only does it help speed calculations, but inter-particle forces at lengths of less than a grid cell are smoothed away . Therefore as two simulation particles approach zero separation the force between them goes to zero instead of infinity, and so coulomb singularities are avoided. Effectively the grid gives the particles a finite size, reducing short-range forces which would otherwise be artificially enhanced by the small numbers of simulation particles. For this reason simulation particles are also known as *clouds* since they have finite size, and can pass through each other.

The numerical scaling factors in the simulation: the time-step Δ_t, the grid-cell width Δ_r, the number of real particles per simulation particle χ, are critical in determining the physical accuracy of the simulation. The choice of these parameters is a balance between the conflicting demands of accuracy (small values), and reasonable run times (large values). Generally the compromise is to chose values which are as large as possible without introducing numerical errors/instabilities. Some of these errors are easy to detect

since they grow exponentially, and cause rapid failure of the simulation, while others are more subtle. Appropriate values for these parameters are determined in Section 2.6.

Particles have a radial position, a radial velocity and a perpendicular energy (which represents the two perpendicular velocity components) and so this code is commonly known as a 1-D spatial, $2\ 1/2-D$ velocity code. In the spherical system it is important to keep track of the perpendicular energy since this has an affect on the radial acceleration due to conservation of angular momentum (see Section 2.4.1). The fields only accelerate particles in the radial direction and so for collisionless particles it is assumed that the particle motion remains directed along the radius and the perpendicular energy can be ignored. However if particles make collisions with non-zero scattering angles then a substantial proportion of the particle energy can be transferred to the perpendicular direction and so for particles which make collisions conservation of angular momentum is important. The transfer of energy into the perpendicular direction tends to reduce motion in the radial direction, meaning that electrons can be trapped by lower potentials and the ions take longer diffuse across the plasma.

Two types of gas are modelled in these simulations: atomic hydrogen and argon. Hydrogen is used for the bulk of the simulations, since it has the lowest ion mass, which reduces the running time, and there are also many theoretical and computational models available for comparison. Simulations using argon are compared with actual experimental data. Positive ions and electrons, which are modelled explicitly as particles, can make collisions with the background gas, which is assumed to form a uniform, constant temperature distribution throughout the reactor volume . As the simulation is designed to model simulations which are only a few percent ionised collisions between charged particles are assumed to be unimportant. Furthermore, in order to keep the simulations simple, hydrogen plasmas are assumed to consist only of atomic hydrogen ions and atoms – this is consistent with assumptions used for analytic models. Collisions are included in the simulation using Monte-Carlo techniques, as described in Section 2.5. For most of the results presented in this thesis the cross-sections have been modelled empirically by fitting curves to experimental data as a function of energy (see Appendix A), particles which make a collision are then scattered isotropically. Isotropic electron collisions are included for all runs; while hydrogen ions can either be collisionless or make isotropic elastic scattering and charge exchange collisions. Argon ions can make anisotropically scattered collisions, using an explicit model of the differential $Ar\text{-}Ar^{+}$ cross-section, the derivation of which is described in Chapter 5. This is used to determine the effect of anisotropic scattering on the ion energy distribution and the results are given in Chapter 6.

In order to start a run particles are loaded uniformly across the length of the plasma with thermal velocities. The simulation is then run for an arbitrary number of time-steps, after which diagnostics are printed out. Runs are continued until the plasma reaches steady-state, at which point data is collected and analysed. The plasma is said to be at

steady-state when quantities such as the plasma density, particle kinetic energies and the potential distribution across the plasma remains constant to within a few percent. Checks have been made on the stability of the plasma over long run-times, and on the consistency of convergence from densities above and below the steady-state value. On average systems in which ions are collisionless reach steady-state at roughly 50 ms of simulation time for hydrogen and 300 ms for argon; systems which include ion collisions can take up to 5 or 6 times longer. Obviously the actual run times are going to depend on the speed of the computer, but, as an indicator, a collisionless hydrogen run takes approximately 36 hours on a DEC alpha to come to equilibrium and an argon simulation with differential ion collisions can take several days.

Sections 2.1 – 2.3 review the basic PIC methods for interpolating from particle positions to the mathematical grid (and vice versa), and the potential and field calculations. Particle motion and time-stepping is described in Section 2.4. The Monte Carlo collision methods, which determine collisions between charged particles and neutrals, are detailed in Section 2.5. In Section 2.6 the numerical limitations of finite difference methods, and the consequent restrictions on simulation parameters, are discussed. A listing of the parameter range used in the hydrogen simulations is given in Section 2.7.

2.1 Charge density assignment

In order to calculate the potential at each point across the plasma, the charge density at each grid point must be known. Particles can be anywhere within a cell, where a cell is defined as the region with boundaries $\pm\Delta_r/2$ from a grid point (i.e. the i^{th} cell has boundaries at i - 1/2 and i + 1/2), so a way of assigning charge from the particle position to the grid point must be determined. For zero order weighting, also known as nearest grid point (NGP) weighting, the number of particles within a cell is counted and their total charge is assigned to the grid point. This effectively gives the particle a rectangular 'shape' with a length equal to the cell width. The NGP method has the benefit of being simple to implement and fast; but has the drawback that as the particle moves across cell boundaries large density fluctuations are experienced at the grid points. This can result in unexceptable noise in the field calculations, since each particle is really a "superparticle" and represents ~10^{10} real particles. This can lead to numerical errors, such as non-physical heating of the electrons, as discussed in Section 2.6.3.

First order weighting is one degree more complicated than the NGP method and involves interpolation of the charge from the particle position to the two closest grid points, which helps smooth the density fluctuations. Higher order weighting (for example using a gaussian weighting over 3 or 4 grid points) further reduces the noise, but takes

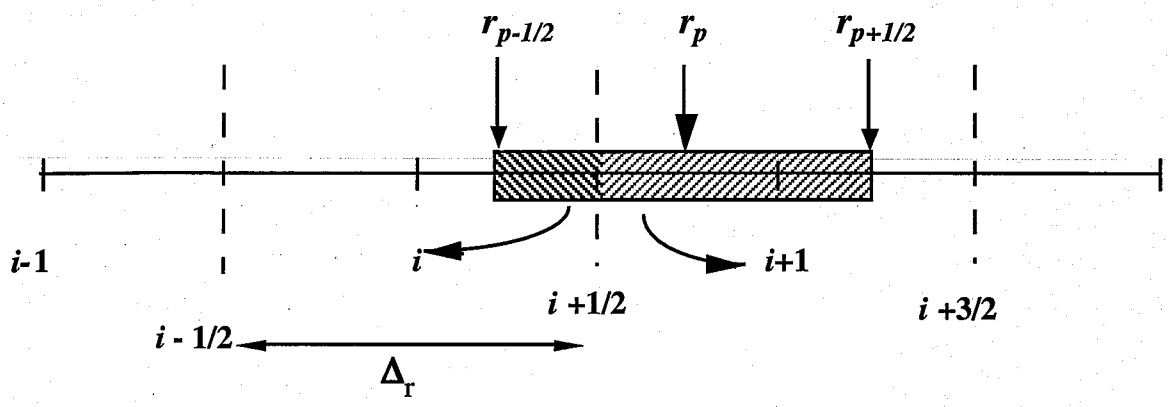

Figure 2.2 Allocation of charge from particle at position r_p to two nearest grid points, i and i+1 using volume weighting

much longer to compute. Speed is one of the major limiting factors of PIC simulations and since the increase in accurately determining the density tends to be hidden by other sources of numerical noise only first order weighting is used for these simulations. In spherical geometry the volume of each cell is proportional to the radius cubed so a first order volume ratio method is used, in which the fraction of the particle charge assigned to the grid point is determined by the volume fraction of the particle within the corresponding cell. Figure 2.2 presents a visual representation of this technique. For computational simplicity the particle width is chosen to be equal to the cell width. First order volume weighting of charge from a particle at position r_p to cell boundary $i+1$ is therefore given by

$$\text{Fraction of charge on } i+1^{\text{th}} \text{ grid point} = \frac{\text{volume of particle in } i+1^{th} \text{ cell}}{\text{total volume of particle}} = \frac{\left(r_{p+\frac{1}{2}}^{3} - r_{i+\frac{1}{2}}^{3}\right)}{\left(r_{p+\frac{1}{2}}^{3} - r_{p-\frac{1}{2}}^{3}\right)}$$

and the fraction of charge weighted to the i^{th} cell boundary is therefore 1 - fraction at $i+1^{\text{th}}$ grid point. Charge-assignation is one of the most time-consuming processes in the simulation, since each particle must be allocated to the appropriate grid points which involves around 10,000 – 50,000 sets of calculations each time-step.

2.2 Potential Calculation

Having determined the densities at each grid point the next step is to determine the potentials using Poisson's equation. In spherical coordinates this has the form

$$\nabla^2\phi(r,\varphi,\theta) = -\frac{\rho(r,\varphi,\theta)}{\varepsilon_o}, \qquad (2.1)$$

where ϕ is the potential, ρ the charge density, ε_O the vacuum permittivity, r the radial coordinate and θ and φ are the angular coordinates. Since the plasma properties are assumed to vary only in the radial direction, equation (2.1) can then be written with ϕ and ρ in terms of r only:

$$\frac{\partial^2\phi(r)}{\partial r^2} + \frac{2}{r}\frac{\partial\phi(r)}{\partial r} = -\frac{\rho(r)}{\varepsilon_o}. \qquad (2.2)$$

To solve this equation it is necessary to obtain boundary conditions at the electrodes. The potential at the outer electrode is defined to be zero, and the electric field at the inner electrode can be determined using Gauss' Law, and so the the potential gradient at this position can be determined:

$$\int_{\text{live electrode}} E.dS = \int_V \frac{\rho}{\varepsilon_o}.dV + \frac{A_a\sigma_a}{\varepsilon_o} \quad ,$$

$$E(r_a + \Delta_r/2)A_a = \frac{1}{\varepsilon_o}\left(\rho(r_a+\Delta_r)\frac{4}{3}\pi\left[(r_a+\Delta_r)^3 - r_a^3\right] + A_a\sigma_a\right) \quad , \qquad (2.3)\ (a)$$

$$\text{where} \quad E(r_a + \Delta_r/2) = \frac{\phi(r_a) - \phi(r_a+\Delta_r)}{\Delta_r} \quad ;$$

and

$$\phi(r_b,t) = 0 \quad , \qquad (2.3)\ (b)$$

where E is the electric field, r_a and r_b the radii of the inner and outer electrodes, Δ_r is the width of one grid-cell, σ_a is the charge density on the inner electrode and A_a is its area.

In finite difference form the radius is written using the notation $r_i = r_o + i\Delta_r$, where $r_o = r_a$. Hence, variables which depend on r have the form $\phi_i = \phi(r_i) = \phi(r_o + i\Delta_r)$. Re-written in finite difference form (2.2) has the is given by

$$\left(1 + \frac{\Delta_r}{r_i}\right)\phi_{i+1} - 2\phi_i + \left(1 - \frac{\Delta_r}{r_i}\right)\phi_{i-1} = -\frac{\Delta_r^2\rho_i}{\varepsilon_o}, \qquad (2.4)$$

where $i = 1, N+1$ (N = number of cells).

In order to reduce the computational cost of multiplying by constants such as the electron mass and Δ_r at each step variables are normalised internally to the simulation. The normalised position, potential and density are given by

$$\tilde{r} = \frac{r}{\Delta_r}, \qquad \tilde{\phi} = \frac{m}{e}\left(\frac{\Delta_t}{\Delta_r}\right)^2 \phi, \qquad \tilde{\rho} = \frac{\frac{4}{3}\pi\Delta_r^{\,3}}{e\chi}\rho, \tag{2.5}$$

where Δ_t is the time-step and χ is the number of real particles per simulation particle. Re-writing (2.4) in terms of the normalised variables gives

$$\left(1+\frac{1}{\tilde{r}_i}\right)\tilde{\phi}_{i+1} - 2\tilde{\phi}_i + \left(1-\frac{1}{\tilde{r}_i}\right)\tilde{\phi}_{i-1} = -\rho_N\tilde{\rho}_i, \tag{2.6}$$

where

$$\rho_N = \frac{e\chi}{4/3\pi\Delta_r^{\,3}}\frac{e}{m}\Delta_t^{\,2}.$$

Converting equation (2.3)(a) to finite difference form, the boundary condition at the live electrode is given by

$$4\pi r_{1/2}^{\,2}E_{1/2} = \frac{1}{\varepsilon_o}\left\{4\pi r_o^{\,2}\sigma_a + \frac{4}{3}\pi\left[\left(r_o+\Delta_r/2\right)^3 - r_0^{\,3}\right]\rho_1\right\},$$

$$\text{i.e.} \qquad E_{1/2} = \frac{r_0^{\,2}}{r_{1/2}^{\,2}\varepsilon_o}\left\{r_o^{\,2}\sigma_a + \frac{1}{3}\rho_1\left[\frac{3}{2}r_0\Delta_r\left(r_o+\Delta_r/2\right)\right]\right\}, \tag{2.7}$$

$$\text{giving} \quad \frac{\phi_o-\phi_1}{\Delta_r} = \frac{1}{\varepsilon_o}\left(\frac{r_0}{r_{1/2}}\right)\left\{\left(\frac{r_0}{r_{1/2}}\right)\sigma_a + \frac{\rho_1}{2}\Delta_r\right\},$$

where $r_o = r_a$ and $r_{1/2} = r_a + \Delta_r/2$. Note that the boundary of the first cell is $\Delta_r/2$ from the electrode, and so the cell is only 1/2 the volume of other cells in the body of the simulation (the same applies to the last cell adjacent to grounded electrode). Note also that while the densities and potentials are known at the grid points, the electric field is known half way between them (this is due to using leap-frog difference methods to determine the electric fields from the potentials), hence the boundary electric field is actually determined at $r_{1/2}$. This is shown schematically in Figure 2.3

By rewriting equation (2.7) and converting to normalised coordinates the potential on the live electrode can be written as

$$\tilde{\phi}_o = \tilde{\phi}_1 + \rho_N\left(\frac{\tilde{r}_o}{\tilde{r}_{1/2}}\right)\left\{\frac{\tilde{\rho}_1}{2} + \frac{\tilde{\sigma}_a}{3}\left(\frac{\tilde{r}_o}{\tilde{r}_{1/2}}\right)\right\}. \tag{2.8}$$

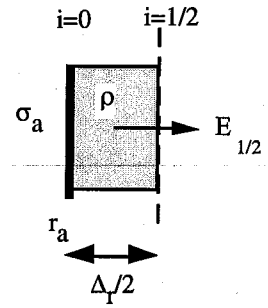

Figure 2.3 Schematic of the electric field $E_{1/2}$ at the boundary of the first cell

Assuming that all variables are in normalised form the tilde can be dropped. Equation (2.6) can be written together with the boundary conditions in (2.8) and (2.3) (b) in a general matrix form, which can then be solved using standard numerical tridiangonal techniques to determine the potentials.

$$\begin{pmatrix} b_o & a_o & & & \\ c_1 & b_1 & a_1 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{N-2} & b_{N-2} & a_{N-2} \\ & & & c_{N-1} & b_{N-1} \end{pmatrix} \begin{pmatrix} \phi_o \\ \phi_1 \\ \vdots \\ \phi_{N-2} \\ \phi_{N-1} \end{pmatrix} = - \begin{pmatrix} d_o \\ d_1 \\ \vdots \\ d_{N-2} \\ d_{N-1} \end{pmatrix}, \qquad (2.9)$$

where

$$a_O = 1, \qquad b_O = 1, \qquad c_O = 0, \qquad d_o = \rho_N \left(\frac{r_o}{r_{1/2}} \right) \left(\frac{\rho_1}{2} + \frac{\sigma_a}{3} \left(\frac{r_o}{r_{1/2}} \right) \right)$$

and for $i = 1, ..., NC\text{-}1$

$$a_i = (1 + 1/r_i), \qquad b_i = -2, \qquad c_i = (1 - 1/r_i), \qquad d_i = \rho_N \rho_i$$

2.3 External Circuit

In order to more realistically model the reactor system an external circuit was included in the plasma simulation, as shown in Figure 2.1. The circuit included a capacitor and (optionally) a resistor, although the latter was not used in the simulation results presented here. The presence of the external blocking capacitor between the rf source and the plasma means that no net current can flow in the circuit. The charge flux to the powered and grounded surfaces must therefore be equal, and furthermore electron and ion losses to each electrode must balance over an rf cycle, or the electrode will become

charged. During the initial time-steps of the simulation (corresponding to breakdown conditions) more electrons escape from the plasma than ions, due to their greater mobility, until the development of the sheaths prevent electrons from reaching the electrodes. At the live electrode this causes the external capacitor to charge up negatively, resulting in an average dc voltage at the electrode known as the bias voltage.

Modelling the external circuit introduces an extra degree of complexity determining the live electrode potential, since the current flowing through the circuit must now be taken into account as well as the flux of plasma particles to the electrodes. The circuit model presented here follows similar methods to those presented in Lawson (1989) and Birdsall and Langdon (1985), page 409. The finite difference form of the circuit equation must be chosen carefully, taking into account the circuit elements included in the simulation, since a circuit containing both inductors and capacitors can lead to problems with numerical instability resulting from circuit resonances. To avoid this at least second order accuracy in time is required for the finite difference equations used to model the circuit. However since an external inductor was not required for this work the equations were simplified significantly to reduce the computational time taken to run the simulation. The following equations model an external circuit with a capacitor (and a resistor).

The charge on the capacitor, Q, can be advanced in time using a finite difference form of Kirschof's voltage law

$$R\frac{\left(Q^{n+1}-Q^{n}\right)}{\Delta t}+\frac{Q^{n}}{C}-V^{n+1}=V_{rf}{}^{n+1} \qquad (2.10)$$

where Q is the charge on the external capacitor, R is the external resistor, C is the external capacitor, V is the voltage on the live electrode, and V_{rf} is the applied rf voltage. The superscript n refers to the variable at time, t_n, and n+1 is one time-step advanced.

When solving (2.10) for time-step t_{n+1} it is necessary to determine two unknowns: Q^{n+1} and V^{n+1}, hence another equation is required. To determine the second equation a relaxation method is used – in the first step the potentials in the plasma due to the distribution of charged particles is calculated while holding the circuit current fixed, and then in the next step the change in potential, due to the flow of charge through the circuit, is determined while holding the plasma potentials fixed. The plasma potentials are then adjusted to take into account the new potential on the electrode.

The second equation required for the potential calculation can be determined from the system capacitance $V = Q/C$, assuming that the dielectric constant of the plasma is close to one. The potential on the live electrode at t_{n+1}, after moving the plasma particles but before including the circuit current, $\hat{V}^{n+1}$, is therefore given by

$$\hat{V}^{n+1} = \frac{A\sigma^n + e(n_i - n_e)}{C_r} = \frac{A\hat{\sigma}^{n+1}}{C_r}, \qquad (2.11)$$

where σ is the charge density on live electrode (n.b. a is dropped for convenience here) andn_i and n_e are the number of ions and electrons which hit the electrode in the period [t_n, t_{n+1}]. Note that ^ is used to refer to quantities that have been calculated *after* the plasma calculations, but *before* the circuit calculations and hence are intermediate values for t_{n+1}. C_r is the capacitance of the reactor system, which for spherical geometry is given by

$$C_r = \frac{4\pi\varepsilon_o r_a r_b}{r_b - r_a}, \qquad (2.12)$$

where r_a and r_b are the inner and outer electrode radii. Holding the plasma particles fixed, the flow of charge in the circuit determines the revised value of the live potential at t_{n+1}

$$V^{n+1} = \frac{A\hat{\sigma}^{n+1} - (Q^{n+1} - Q^n)}{C_r}, \qquad (2.13)$$

where Q is the charge stored in the circuit. The change in potential at the live electrode is therefore given by the difference between (2.11) and (2.13)

$$V^{n+1} - \hat{V}^{n+1} = -\frac{Q^{n+1} - Q^n}{C_r}. \qquad (2.14)$$

Substituting (2.14) into (2.10) gives an equation for Q_{n+1}

$$R\frac{Q^{n+1} - Q^n}{\Delta t} + \frac{Q^{n+1}}{C_{tot}} = \hat{V}^{n+1} + V_{rf}{}^{n+1} + \frac{Q^n}{C_r}, \qquad (2.15)$$

where C_{tot} is the total capacitance, i.e., $1/C_{tot} = 1/C_r + 1/C$.

Equation (2.15) is only first-order accurate in time, however it can be made second-order accurate by doing a time-expansion on all the variables and including extra terms. This has the advantage of making the circuit solution much more numerically stable in time, otherwise oscillations between the plasma and circuit equations can lead to unphysical effects, which can eventually cause catastrophic failure of the simulation. This calculation is fully derived in Lawson (1989) and so the details are not presented here. The second-order version of (2.15) is:

$$R_m \frac{Q^{n+1} - Q^n}{\Delta t} + \frac{Q^{n+1}}{C_{tot}} = \frac{\hat{V}^{n+1} + V^n}{2} + \frac{V_{rf}{}^{n+1} + V_{rf}{}^n}{2} + \frac{Q^n}{C_r}, \qquad (2.16)$$

where R_m is a modified version of the resistance, used to force an exponential decay time for the external capacitor, and is calculated using

$$1 + \frac{\Delta_t}{R_m C_{tot}} = \exp\left(\frac{\Delta_t}{R C_{tot}}\right) \qquad (2.17)$$

Finally the potentials initially calculated within the plasma must be modified due to the new potential at the electrode . The modification can be determined using Laplace's equation and superposition of the potentials (since potential is a scalar quantity). In terms of r Laplace's equation is given by

$$\frac{\partial^2 \phi_L}{\partial r^2} + \frac{2}{r}\frac{\partial \phi_L}{\partial r} = 0, \qquad (2.18)$$

where, ϕ_L is the potential correction, which has the general solution

$$\phi_L = \frac{a}{r} + b$$

where $a = \frac{V_{n+1} r_a r_b}{r_b - r_a}$ and $b = -\frac{V_{n+1} r_a}{r_b - r_a}$.

Therefore the new potentials are given by $\phi(r) = \phi'(r) + \phi_L(r)$ where ϕ' are the potentials calculated from the matrix in equation (2.9).

A second-order time derivation for the full RCL circuit has been derived by Verboncoeur *et al* (1993), for models in which it is desirable to model an external circuit which includes both a capacitor and an inductor.

2.4 Electric Fields and Particle Motion

Changes in particle motion are a result of the action of the electric field on the (charged) particle. New velocities for the particles are therefore calculated each time step using the electric field at the particle position. The finite difference form for the electric field calculations is simply

$$E_{i+1/2} = -\frac{\phi_{i+1} - \phi_i}{\Delta_r}. \qquad (2.20)$$

Since a two-point finite difference scheme is used, the fields are determined at positions half-way between grid points. Care must be taken when calculating the field at the boundaries and in interpolating the field from the grid to the particles. The same weighting method described in Section 2.1 is used to determine the fields at the particle positions – this stops particle self-forces from occurring, in which particles move in the fields they themselves have produced. The force on the particle, due to the electric field, is

$$F_m = m\frac{\partial v_m}{\partial t} = eE_m, \qquad (2.21)$$

where the subscript m refers to the quantities relevant to the m^{th} simulation particle.

2.4.1 Perpendicular energy

In the spherical model a particle moving in a straight line past the central electrode will appear to have a radial force acting upon it. This is shown diagrammatically in Figure 2.6 (in two dimensions for simplicity) for the case when no potentials are applied to the electrodes, so there are no electrostatic forces on the particles. Figure 2.6 (a) represents a particle bypassing the electrode, and (b) plots the trajectory in the radial direction showing the action of an apparent force on the particle. This force is actually due to conservation of the particle conservation of momentum in spherical geometry. To account for this in the one-dimensional system, in which particles can only move in the radial direction, the perpendicular energy of the particle (representing the two perpendicular velocities) is determined for each particle and conservation of momentum is used to calculate the magnitude of the pseudo force.

The radial acceleration of the particle due to its perpendicular velocity is determined from the familiar equation for conservation of angular momentum:

$$\frac{\partial v_r}{\partial t} = \frac{v_p^2}{r}, \qquad (2.22)$$

where v_r is the radial velocity, v_p is the (total) perpendicular velocity. From equations (2.21) and (2.22) the total force on the m^{th} particle, due to both the electric field and its angular momentum, is

$$F_m = eE_m + \frac{\left[v_p^{\,2}\right]_m}{r_m}. \qquad (2.23)$$

From Newton's equations of motion, the particle's new velocity and position can be calculated at each time-step. Using normalised variables $\tilde{v} = (\Delta_t/\Delta_r)v$, $\tilde{r} = r/\Delta_r$ and $\tilde{E} = (e\Delta_t^{\,2}/m\Delta_r)E$, Newton's equations can be written in finite difference form as

$$\tilde{v}^{n+1/2} = \tilde{v}^{n-1/2} + \tilde{E}^n + \frac{\left[\tilde{v}_p^{\,2}\right]^n}{\tilde{r}^n}, \qquad (2.24)\ (a)$$

$$\tilde{r}^{n+1} = \tilde{r}^n + \tilde{v}^{n+1/2}, \qquad (2.24)\ (b)$$

$$\left[\tilde{v}_p\right]^{n+1} r^{n+1} = \left[\tilde{v}_p\right]^n r^n, \qquad (2.24)\ (c)$$

where the superscript n refers to the time-step. The subscript m has been dropped since it is implicit that the set of equations is applied to each particle in the plasma. Note that the positions, electric fields and perpendicular velocities are known at integral time-steps, while the velocities lag 1/2 a time-step behind. This is a finite difference technique known as the leap-frog method, which is used because it is simple to implement, requires minimal storage of variables, and is numerically stable for a wide range of conditions. For example, Birdsall and Langdon (1985, page 56) show that using this scheme to

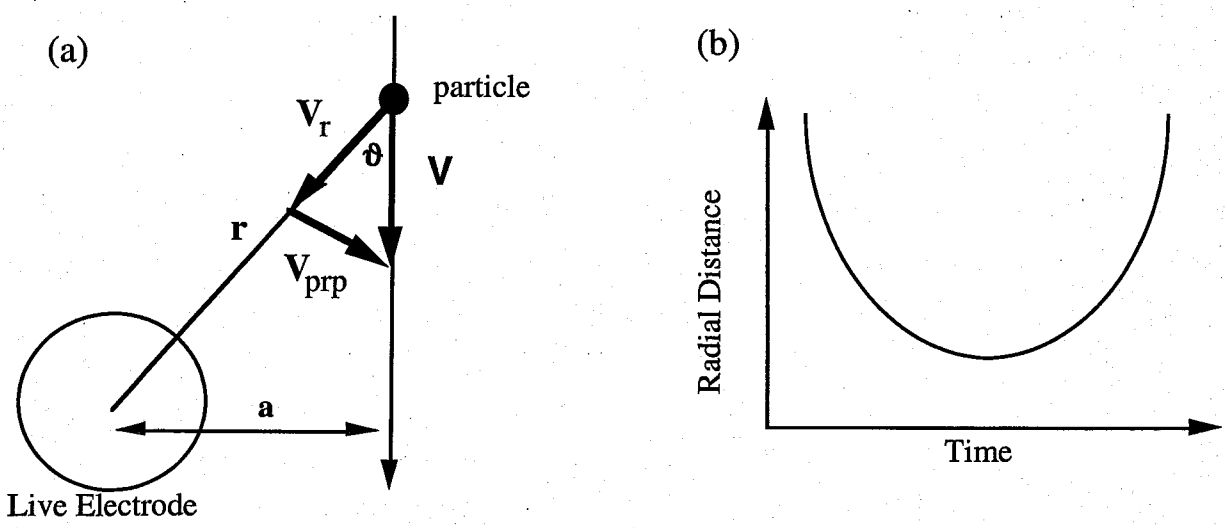

Figure 2.6 Particle trajectory in a spherical system when no potential is applied to the electrode **(a)** Particle motion showing radial and perpendicular velocities **(b)** radial position versus time of a particle passing the central electrode, showing the effect of the "pseudo" force acting in the radial direction.

numerically integrate a harmonic oscillator with frequency ω_o, gives a solution which is exact in amplitude and accurate in phase to order $(\omega_o \Delta_t)^3$ when $\Delta_t \leq 2/\omega_o$. Since the electron plasma frequency, ω_p, is the highest frequency oscillation which needs to be resolved by the simulation the leap-frog scheme should be numerically stable for $\Delta_t \leq 1\,\text{ns}$.

Note also from equation (2.24)(a) that the fields which accelerate the particles are only calculated at the original position of the particle, and no account is taken of changing field gradients as the particle moves. This implicitly assumes that particles will not travel far enough in one time-step for substantial changes in the field to occur. One problem with this assumption is that at certain times during the rf period the motion of the sheath can accelerate electrons in the vicinity of the sheath edge to very large energies (several hundreds of eV). This effect is especially noticeable at the live electrode in systems with very asymmetric geometries, due to the large sheath potentials in these systems. These energetic electrons can travel over several cells during one time-step without taking into account the field gradients which can occur over these distances. Only relatively few electrons attain these energies so it is inefficient to decrease the time-step for the whole ensemble of simulation particles. Hence a simple proceedure is implemented in which fast electrons are picked out and moved by fractions of the time-step with the force on the electron re-calculated for sub-step. Note that fields on the grid are *not* re-calculated for each sub-step, so there is an implicit assumption that only a very small fraction of the total electron density is moved in this manner and hence the fields remain relatively unperturbed. This is a reasonable assumption since generally only a small number of electrons close to the sheath edge are effected.

2.5 Particle collisions

Collisions are included in PIC simulations using Monte Carlo methods, a powerful and widely used computational technique in plasma simulations in which random numbers are used to determine particle behaviour. For this model the null collision technique is used to pick out a particle to test, and then the relative probability of collision is tested against a random number to determine if a collision actually occurs. In this code since there are actually three types of electron collisions and two of ion collisions, this technique is also used to determine the type of collision. The most straightforward way of including collisions involve testing *all* of the particles in the simulation *each* time-step however this would make the process extremely time-consuming processes and so a technique known as the null collision method is used. This is detailed in Section 2.5.1

For these simulations only electron-neutral and ion-neutral collisions are considered – the plasma is weakly ionised, and so the plasma density is low enough that

the probability of collisions between charged particles is considered to be small. This means that small angle Coloumb collisions, which tend to heat the coldest electrons, are not explicitly included in the code, although other authors have done so (see Birdsall (1991)). The technique tends to be computationally time-consuming, and in these simulations the pressure is sufficiently large that coloumb heating is not significant. All of the simulations include isotropic scattering of electrons, using energy dependent total cross-sections. In Chapter 4, results are presented from simulations of atomic hydrogen plasmas, both for the case in which ions are assumed to be collisionless and for the case in which ions make isotropically scattered elastic or charge-exchange collisions with neutrals. In Chapter 5 a detailed model of Ar-Ar^+ differential cross-sections is presented, which uses the interaction potential of the ion-neutral pair. This is then used in argon plasma simulations and results are compared to experimental measurements in Chapter 6. Collisions have an important effect on the particle motion since, especially for isotropic scattering, they tend to lead to transfer of energy from the radial into the perpendicular direction.

In order to select particles to test for a collision event the mean free time between collisions is calculated

$$\tau_c = \frac{1}{\nu_c} = \frac{1}{N_n Q(\varepsilon) v} \qquad (2.25)$$

where ν_c is the collision frequency,v is the particle velocity, $Q(\varepsilon)$ the energy dependent collision cross-section and N_n the neutral gas density. In order to determine which particles make collisions an incremental count is kept as each ion or electron is moved during the time-step until the accumulated time is equal to (or greater than) τ_c and that particle is selected to test for a collision; the count is set back to zero and testing is recommenced. Note that if the cross-section is proportional to v, the particle velocity, then τ_c will be a constant for all energies. This saves a lot of time in testing, since τ_c does not have to be re-calculated for each particle, this leads to the concept of pseudo cross-sections and null collisions.

2.5.1 The null collision method

The process used for determining collisions is described for electrons, since electron collisions are used in all of the simulation runs; a similar method is used for isotropic scattering of the ions. Electrons make three types of collisions – ionisation, excitation, and elastic scattering. Ionisation is the most important process, since in order to sustain the plasma it is necessary to include some source term of charged particles, to

balance those lost to the walls. The main effect of excitation is to cool the electrons; and elastic scattering simply redistributes the velocity. Although electrons are known to scatter anisotropically, especially at high energies, this effect is considered relatively unimportant at the low electron temperatures seen in the simulation – most of the electrons form a background 'sea' of randomly directed thermal electrons – so the scattering angles are assumed to be isotropic

The total collision cross-section, Q_{tot}, can be determined from the sum of the individual cross-sections for elastic scattering, ionisation and the various excitation collisions. For the purpose of this work, only the 1s to 2p excitation cross-section is used. Empirical fitting of curves to experimental data was used to determine equations for energy dependence of the cross-section. Details of the cross-sections and their fitting is given in Appendix A. Having determined the total cross-section by summing over all of the cross-sections for the appropriate energy range, a pseudo cross-section is defined:

$$Q_{pseudo} = \frac{Q_o}{v}, \tag{2.26}$$

where Q_o is a constant, chosen so that Q_{pseudo} is greater than Q_{tot} for all energies (see, for example, Figure B.1 for the e-H cross-sections). Then the time between collisions for the pseudo cross-section will be a constant for all electron velocities $\tau_{pseudo} = N_n Q_o$.

Having picked an electron the relative probability of each type of collision is determined from the ratio of the individual cross-sections to the pseudo cross-section at that energy.

$$Pr_{ion} = \frac{Q_{ion}(\varepsilon)}{Q_{pseudo}(\varepsilon)}, \quad Pr_{excn} = \frac{Q_{excn}(\varepsilon)}{Q_{pseudo}(\varepsilon)}, \quad Pr_{elas} = \frac{Q_{elas}(\varepsilon)}{Q_{pseudo}(\varepsilon)}, \tag{2.27}$$

where Q_{ion}, Q_{excn}, and Q_{elas} are the ionisation, excitation and elastic scattering cross-sections respectively. The collision probabilities are summed, as shown diagrammatically in Figure 2.7, and a random number, $\Re$, uniformly distributed between [0,1], is used to determine which type of collision takes place.

Figure 2.7 demonstrates schematically the relative probabilities of each type of collision and how a random number is used to select which collision will occur. Note that there is a category Pr_{null} – this comes about because the pseudo cross-section is always chosen to be greater than the total cross-section, and so there is a finite probability that no collision will take place. This represents a null collision event, hence the name of this technique. Ionisation and excitation also have conditions on the mimimum energy of the electron, which must be fulfilled before the collision can take place, due to the forms of their cross-sections. When an ionisation event occurs a new ion and electron must be

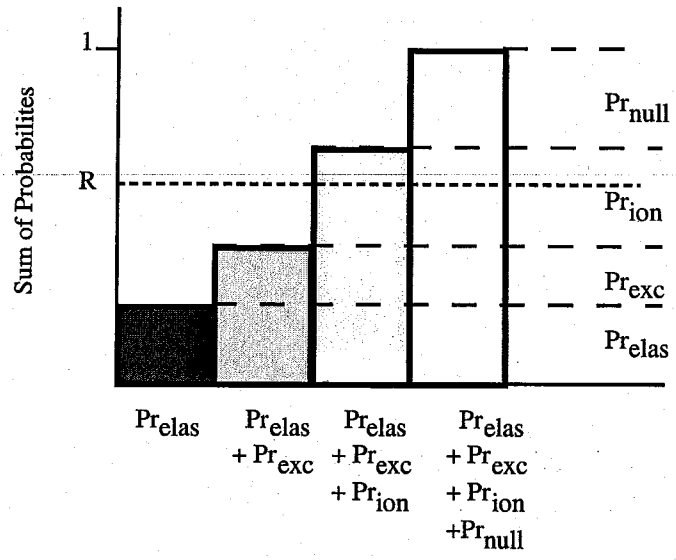

Figure 2.7 Schematic of the probabilities of collision – each collision probability is summed, and then a random number is picked to determine the type of collision that occurs. In this case it is an ionisation collision.

added to the simulation. The velocity for the new ion is picked from a 0.025 eV Maxwellian distribution (since it was previously a background gas atom). The energy of the original electron, minus the ionisation energy, is shared randomly between the old and new electrons, and each of the electrons is isotropically scattered. When an excitation eventoccurs the electron loses the threshold energy and is randomly scattered. For an elastic scattering collision the electron loses a very small amount of energy, due to momentum transfer to the neutral, and is then isotropically scattered.

For isotropic ion collisions the ions to be tested are similarly chosen and a random number is used to determine whether an elastic scattering, a charge-exchange, or a null collision event occurs. If the ion is elastically scattered a scattering angle is chosen using another random number from $\cos\theta = [1 - 2\Re]$. Note that this is the scattering angle in the centre-of-mass frame. The scattering angle is used to find the new parallel velocity and perpendicular energy in the centre-of-mass frame, which are then converted back to lab frame coordinates. The calculations are given in detail in Section 5.2.2. In the event of a charge-exchange collision, the neutral and the ion exchange an electron, resulting in a hot neutral, and a cold ion (this is indistinguishable from a 180° elastic scattering collision), and the new velocity distribution for the ion is again picked from a 0.025 eV Maxwellian distribution.

Finally it must be ensured that the time-step is smaller than the mean time between collisions, otherwise there exists a finite probability that a particle could make more than one collision during a time-step, which is not accounted for in the code. Normally this is not a problem, since the mean free time between collisions for both electrons and ions in

hydrogen at a pressure of 20 mTorr is ~ 10 ns, much larger than the time-step $\Delta_t = 0.1$ ns. However at larger gas pressures the time between collisions is reduced, so at 500 mTorr $\tau_c \approx 0.6$ ns and a smaller time-step would be required. For the hydrogen simulations the maximum pressure used is 100 mTorr.

2.6 Scaling

Something needs to be said about the way the scaling in the plasma works, since this is fundamental in relating simulation plasmas to the 'real world'. In particular the effects of using finite difference techniques on linear wave dispersion and conservation of energy, and how grid cell size determines the stability and accuracy of PIC systems (Langdon and Birdsall (1970), Langdon (1979), Birdsall (1991))

The scaling of variables, such as the grid-cell width, Δ_r, the time-step, Δ_t, and the "superparticle" density, n_o, is very important since these parameters control the numerical accuracy of the simulation. They have to be carefully chosen in order to ensure that the results from the simulation are due to the physics included in the code, rather than numerical effects introduced by approximating the analytic equations with finite difference versions. Obviously the accuracy of the representation will improve as the size of the scaling factors is decreased (i.e. $\frac{\Delta\phi}{\Delta_r} \rightarrow \frac{\delta\phi}{\delta_r}$ as $\Delta_r \rightarrow 0$), but this makes the simulation very long to run – the time taken to run the simulation scales directly with Δ_r and Δ_t and depends roughly on the number of particles squared. One of the major features of PIC is that it explicitly models individual particles and their trajectories. This has the drawback of making the codes computationally expensive, but on the other hand allows determination of macroscopic effects due to individual particle motions. PIC codes are particularly suited to examining sheath phenomena – a region that is not well handled by other numerical methods, such as fluid codes. Hence Δ_t, Δ_r and n_o are chosen to be as small as possible, without introducing errors which are cumulative in time, or which are going to effect the representation of the physics. In the main the appropriateness of these parameters has been checked empirically, by doubling the quantities several times and checking for changes at steady-state conditions. The following sections give a brief summary of the main restrictions and accuracy of the scaling factors.

2.6.1 Grid cell width

The grid cell width Δ_r is generally chosen to be of the order of the debye length. The debye length represents the maximum distance over which the coulomb forces of

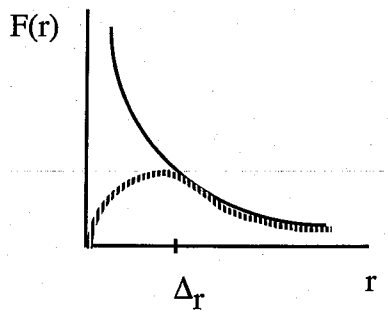

Figure 2.8 Coulomb force for a point charge (solid line), and for a particle of width Δ_r (broken line).

individual particles are important. By setting the cell width equal to the Debye length short-range particle interactions are reduced, these are unimportant to the overall coherent plasma behaviour and otherwise dominate force calculations in the simulation. Effectively what the grid does is to give particles a finite size equal to the cell width, so that as particles move "through" each other the inter-particular force goes to zero, instead of infinity as would be the case for point particles. This is shown schematically in Figure 2.8. For this reason PIC particles are often referred to as *clouds* to differentiate them from point particles. Reducing the cell width makes no change to the final equilibrium the plasma reaches, but increases the number of calculations required.

However, if the cells are made much larger than the Debye length, then important electric field gradients can be ignored, since the fields are only determined at the grid points. Furthermore since the cell widths define the particle size, values which are larger than the debye length lead to very unphysical particles. For a plasma in the density range $10^{14} - 10^{15}$ m^{-3}, with an electron temperature of 3 eV, the debye length is approximately $10^{-3} - 4\text{x}10^{-4}$ m; and so Δ_r is chosen to be $2\text{x}10^{-4}$ m. For the simulation results presented here the electrode separation is 0.2 m and so the simulation typically uses 1000 grid cells.

Although adding a spatial grid to the simulation makes density and force calculations very much more efficient, the grid can also introduce non-physical perturbations to the plasma. For example, with the inclusion of a grid the interaction force, F, between two particles depends on the mean particle position relative to the grid as well as particle separation – spatial invariance is lost due to the presence of the grid. This effect can be mitigated by use of finite sized particles and smoother weighting techniques.

This dependence of the interaction force on the average position of the particles results in non-physical mode coupling. This effect is analysed in detail in Birdsall and Langdon (1985, Chpt 8) and Tajima (1989, Chpt 4). Briefly, mode coupling means that oscillations which differ by a factor of $2\pi/\Delta_r$ are indistinguishable to the simulation; this is shown schematically in Figure 2.9. The effect is known as *aliasing* and is due to the loss of information in going from a continuous system to a spatially discrete system. However,

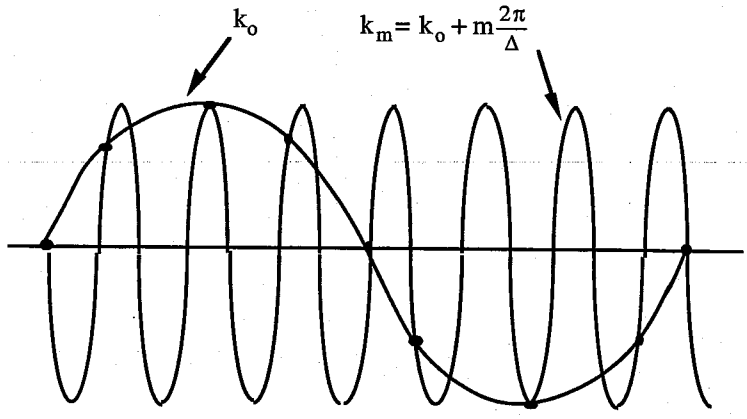

Figure 2.9 Wave-coupling (aliasing) for oscillations with wavenumbers which differ by an integral number of the inverse grid-cell width.

provided the grid cells are less than the smallest wavelength of interest, this coupling will not significantly effect the simulation results.

2.6.2 *Time Steps*

Time-stepping can be quite complicated since the time scales in the plasma typically vary by orders of magnitude. For example, at a densities of 10^{14} m^{-3} the electron plasma period is around 10^{-9} s, while the ions move on time scales of about 10^{-7} s, and the mean free time between collisions is of the order of 10^{-8} s. Generally speaking the electron plasma period is the smallest time-scale to be resolved, although for very high pressures care must be taken to resolve the collision period. Using the electron time-scale for the ions however over-resolves the ion motion and so involves more calculations than strictly necessary. Although it is possible to move the ions and electrons using different time-steps it makes the potential and force calculations very much more complicated, and so the time-step for these simulations is simply chosen to be small (~0.1 ns).

Using a finite time-step Δ_t results in a periodicity in time, similar to the effect of the grid mentioned in the previous section. Again this causes aliasing of oscillations, but now it occurs for those differing by harmonics of $\frac{2\pi}{\Delta_t}$ – effectively the time-step acts like a stroboscope. As before this is not important provided the time-step is chosen to be less than the smallest period of interest in the plasma.

2.6.3 *Superparticle Number*

Numerical heating is mainly an undesirable side-effect of representing the large number of charged particles in a plasma by relatively few "superparticles", although it is also affected by the grid-cell width, time-step and simulation dimensions. Essentially it is non-physical heating of the electrons through potential oscillations which are caused by the

"graininess" of the plasma. Particles moving across cell boundaries cause fluctuations in the density and this noise is then passed on to the potentials and the electric fields. As electrons are very sensitive to fluctuations in the field this can result in an unphysical increase in the electron energy. Ions are much slower to respond to the fields and so they are not affected by transient fluctuations. To a certain extent this noise is simply an ineradicable part of numerical simulations and it can be ignored, provided the fluctuations due to noise are smaller than the amplitude of the applied potentials *and* they do not result in unstable behaviour. If, however, this artificial boosting of the electron energy becomes large enough it can start to cause excess ionisation, leading to an increase in the plasma density, which magnifies the heating, and eventually results in an exponential, runaway increase in the plasma density and finally catastrophic breakdown of the simulation.

A rough calculation of the ratio of the potential fluctuations and the electron temperature is used to determine the conditions for which numerical heating is likely to occur. Since this is only meant to give a feel for the important parameters the derivation is presented for a planar rather than spherical system, which considerably simplifies the equations. The potential due to the fluctuation in space-charge is therefore

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{e}{\varepsilon_o} \Delta\left(n_i - n_e\right). \tag{2.28}$$

If the superparticle number is $\chi = N$ then the density can only be known to a factor of 1/N, so the fluctuation in the density caused by one particle is given by n_i/N (assuming that $n_i \approx n_e$), that is $\Delta\left(n_i - n_e\right) = n_i/N$. Assuming that in the case of no external voltages the plasma potential has a roughly parabolic form (see Figure 2.10), then the equation for the potential distribution is given by

$$\phi(x) = \phi_{\max} x(L-x)/(L/2)^2. \tag{2.29}$$

Differentiating (2.29) twice, and equating to (2.28) gives

$$\phi'' = \frac{8\phi_{max}}{L^2} = \frac{e}{\varepsilon_o}\frac{n_i}{N}. \tag{2.30}$$

Rearranging (2.30) and multiplying both sides by e/kT_e, to determine the ratio of the maximum potential fluctuation to the electron temperature, gives

$$\frac{e\phi_{\max}}{kT_e} = \frac{1}{8}\left(\frac{e^2 n_i}{\varepsilon_o kT_e}\right)\left(\frac{L^2}{N}\right) = \frac{1}{8N}\left(\frac{L}{\lambda_D}\right)^2 \tag{2.31}$$

where $\lambda_D = \sqrt{e^2 n_i/\varepsilon_o kT_e}$ is the Debye length.

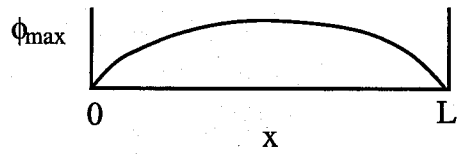

Figure 2.10 Form of space-charge potential in a system with no applied potentials.

Hence for large systems and high densities (which generally means large N) care must be taken to ensure that the density fluctuations do not seriously affect the electron temperature. This can be accomplished by running with a large number of particles (i.e., reducing χ), although it will result in much longer run times. In some cases reducing the grid cell width or the time-step can also help, since the density is then better resolved.

Another problem resulting from using a single massive particle to represent χ real particles is loss of resolution in regions of low density, such as the sheaths. Obviously regions with densities less than χ cannot be adequately represented. Porteous *et al* (1994) circumvented this problem by using a variable value of χ, so that in regions of lower density superparticles are smaller, leading to improved resolution. The main effect on the simulations presented in this thesis would lie in overestimating the number of hot electrons. In low pressure, rf systems the plasma is maintained through ionising collisions made by a relatively small number of hot electrons created by interaction with the moving sheath edge. Superparticles could result in a slight overproduction of hot electrons, giving plasma densities which are larger than would be otherwise predicted. However, this effect has not been observed so far, and even if it occurs is expected to be very small.

2.7 Run parameters

This section details the range of parameters used in the different simulations. A large number of system geometries have been explored in order to determine the effect of asymmetry on the plasma parameters, since this is of particular interest in the spherical system. In all cases the electrode radii were altered and the electrode separation kept constant; the asymmetry of the system was measured using the ratio of the electrode areas, i.e., $\alpha = r_b^2/r_a^2$. Simulation results are also presented for a wide range of applied voltage, V_{rf}, and background gas pressures, p, as well as for a few different source frequencies, f_{rf}. This was carried out in order to observe the plasma behaviour for a broad spectrum of physical conditions. The values of the parameters used in the simulations are shown in Figure 2.1.

A number of analytic and numerical models (Horrowitz(1983), Lieberman (1989), Alves *et al* (1991), Raizer and Shneider (1992)) of asymmetric systems have been used to examine the effect of changing the ratio of the electrode areas on the ratio of the sheath potentials, and on the formation of a self-bias voltage. However, to the author's

knowledge, there has been no previous systematic study on the effect of asymmetric geometry on plasma parameters such as electron temperature, power deposition, density profiles, currents and so on. This study concentrates on the effect of changing the electrode area ratio on the plasma behaviour, in order to determine some analytic and empirical scaling laws for asymmetric discharges, and to determine the degree of agreement, or otherwise, with previously derived laws. The effect of varying the applied voltage in an asymmetric system was also examined in detail, in order to compare to previously PIC results in planar systems and to determine whether planar derivations still hold when extrapolated to spherical geometry.

It is not possible to run enough simulations to vary every single different parameter individually (it would require more than 2.4×10^{18} runs), and so a set of baseline parameters were chosen, shown as shaded values in Table 2.1. For each set of simulations three of the baseline values were held constant while the fourth was varied over its parameter range. Hence when studying, for example, the effect of changing area ratio seven simulations were run with values of α between 1 and 15, while the other parameters were kept at $V_{rf} = 1$ kV, $f_{rf} = 10$ MHz and $p = 20$ mTorr.

This method does have a drawback, since the plasma characteristics depend on *all* of the parameters concurrently, while this method assumes that each parameter can be varied independently. For example, Figure 48 shows the dependance of bias voltage on area ratio, but this is only valid for one particular applied voltage, and for other voltages the dependance on α could be quite different (e.g., decreasing instead of increasing). It must therefore be assumed that the plasma would behave consistently for other values of the baseline parameters.

Area Ratio, α	Voltage, V_{rf} (V)	Pressure, p (mTorr)	Frequency, f_{rf} (MHz)
1	100	10	5
1.5	200	20	10
2	500	50	20
4	1000	100	
6	2000		
9	5000		
15			

Table 2.1 Range of parameters used in the simulation. The shaded values are those used as "baseline" parameters, e.g., when looking at variation in voltage the other parameters used are $\alpha = 6$, $p = 20$ mTorr and $f_{rf} = 10$ MHz

Chapter 3

Analytic model of the simulation

This chapter presents analytic models for the bulk and the sheaths, results from which are compared to the PIC simulation. Results from other numerical codes are also compared to the PIC and analytic results. As mentioned in Chapter 1 most of the PIC simulations were run using atomic hydrogen gas with collisional electrons and collisionless ions. These conditions were chosen since cross-sections for hydrogen are well known, and having the lightest ion mass hydrogen reaches steady-state conditions quickly. These simulations were primarily designed to determine the effect of the "external" or applied parameters – such as area ratio, voltage, pressure and frequency (as detailed in Table 2.1) – on the "internal" plasma characteristics, that is the electron temperature, the average potential and density distributions, ion currents, sheath widths etc.

In numerical work the number of diagnostics included is essentially limited only by time constraints for computation – this potentially means there is a vast body of information about the plasma for each different set of conditions. The problem is then to organise this information into some meaningful form, such as scaling laws relating the external input parameters to the internal plasma parameters, in order to further understand the physical processes driving the plasma. Furthermore it is desirable to check the validity of the PIC model, by determining whether the physical picture derived from it accords with other numerical and analytic models. Comparison to experimental results is not really possible for the parameters shown in Table 2.1, since these conditions were not chosen to represent a real physical system, rather they represent a simplified view for comparison with one dimensional models. Hence the analytic model presented here was designed to incorporate the physical features of the PIC code – spherical geometry, collisionless ions - and by tailoring the analytic model to fit the simulation results an understanding of the important physics in the simulation and the consequent effect on the macroscopic behaviour of the plasma is obtained.

Low pressure capacitively coupled rf systems have been quite intensively modelled over the past twenty years or so, because of their technical applications in the materials processing industry. A number of analytic models, including assumptions and