

Particle simulation of plasmas: review and advances

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2005 Plasma Phys. Control. Fusion 47 A231

(<http://iopscience.iop.org/0741-3335/47/5A/017>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 198.125.177.150

The article was downloaded on 03/03/2011 at 18:52

Please note that [terms and conditions apply](#).

Particle simulation of plasmas: review and advances

J P Verboncoeur

Department of Nuclear Engineering, University of California, Berkeley, CA 94720-1730, USA

E-mail: johnv@nuc.berkeley.edu

Received 29 October 2004

Published 19 April 2005

Online at stacks.iop.org/PPCF/47/A231

Abstract

Particle simulation of plasmas, employed since the 1960s, provides a self-consistent, fully kinetic representation of general plasmas. Early incarnations looked for fundamental plasma effects in one-dimensional systems with $\sim 10^2$ – 10^3 particles in periodic electrostatic systems on computers with $\lesssim 100$ kB memory. Recent advances model boundary conditions, such as external circuits to wave launchers, collisions and effects of particle–surface impact, all in fully relativistic three-dimensional electromagnetic systems using $\sim 10^6$ – 10^{10} particles on massively parallel computers. While particle codes still enjoy prominence in a number of basic physics areas, they are now often used for engineering devices as well.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

This paper is intended to provide a brief overview of the history and recent advances in the field of particle simulation of plasmas. This work represents over thirty-five years of experience both within the simulation community connected in various ways to the University of California at Berkeley Plasma Theory and Simulation Group (PTSG) and those outside it. Within the constraints of the present format, it is not possible to give even a shallow treatment of every important area in the field, so omission should not be construed as a commentary on the relative importance but rather a consequence of the author's familiarity with the model in question. Furthermore, many schemes that push the envelope of the particle method are beyond the scope of what is intended to be a treatment of the fundamentals. Finally, once again hiding behind the limits of the format, it is necessary to leave many details to the literature.

1.1. Motivation

Perhaps one of the strongest motivations for pursuing plasma physics is the estimate that 99% of the matter in the universe is in the plasma state. Here, we will use the term plasma inclusively, to encompass beams and other non-neutral collections of charged particles, independent of

density, temperature and Debye length. Most plasmas have nonlinear effects of importance. A modest fraction of plasmas has a sufficiently low collision frequency and/or are acted upon by sufficiently strong forces that kinetic effects also play a role. Many plasmas, especially terrestrial ones, have boundaries that play a prominent role, and also give rise to nonlinear and kinetic effects. Collective effects also play an important role in many plasmas of interest.

The particle-in-cell (PIC) method allows the statistical representation of general distribution functions in phase space. The use of fundamental equations often retains the full nonlinear effects, and space charge and other collective effects can be included self-consistently by coupling charged particles to the field equations via the source terms. The particle treatment allows the incorporation of relativistic effects. In general, the PIC method employs the fundamental equations without much approximation, allowing it to retain most of the physics.

Despite having many advantages, the PIC model also has a number of weaknesses. Perhaps the most quickly encountered of these is computational efficiency. This is a consequence of the statistical model in which numerical fluctuations converge as $N^{-1/2}$ for N particles; the PIC scheme and some modifications can reduce the constant but not the scaling. An associated problem is the difficulty in resolving the tail of the distribution, which is often poorly populated by statistical methods. It is also challenging to model large ranges of timescales, as short timescales require small time steps while long timescales require running many time steps. Similarly, large ranges of space scales present similar difficulties for the mesh size. Finally, the PIC method requires significant memory and processor resources, and for the foreseeable future this will remain the case. Fortunately, the boundaries of what is possible are always advancing by virtue of Moore's law and new algorithms.

1.2. A brief history

The roots of the PIC method go back to self-consistent calculations performed by Buneman [6] and Dawson [8] in the late 1950s. In these basic physics models, space charge forces were included via direct solution of Coulomb's law, and charged particle trajectories were computed in periodic systems. Nevertheless, the power of computational plasma physics was established.

In the next decade, significant advances were made. Early models calculated Coulomb's law for each particle, an N^2 operation for N particles. Collectively, the growing computational plasma community recognized the need for a model with better scaling in N , and the particle-mesh methods were devised by imposing a computational mesh on which to compute Poisson's equation. Depending on the weighting schemes, these methods were referred to as *cloud-in-cell* or *particle-in-cell*, and were capable of simulating systems with $\sim 10^3$ – 10^4 particles in periodic systems, generally in one dimension. These schemes included collective effects, especially space charge forces, and were applied with great success to basic unbounded plasma problems. In 1964, Dawson [9] demonstrated the effects of Landau damping of electrostatic waves predicted by theory but not yet observed experimentally, further boosting the belief in computational plasmas. Many others made notable discoveries or validations using the PIC scheme.

The PIC scheme was formalized and codified during the 1970s, although the classic texts were published in the following decade by Birdsall and Langdon [1] and Hockney and Eastwood [13]. These texts remain pre-eminent to the present. Much of the theory for the PIC method was worked out at that time, including the theory of finite time and space discretizations; [1] remains the most complete treatment. Electromagnetic schemes for propagating waves were also developed in this period.

In the 1980s, it was recognized that boundaries played an important and not well-understood role in plasmas. The Plasma Device Workshop, convened in Berkeley by

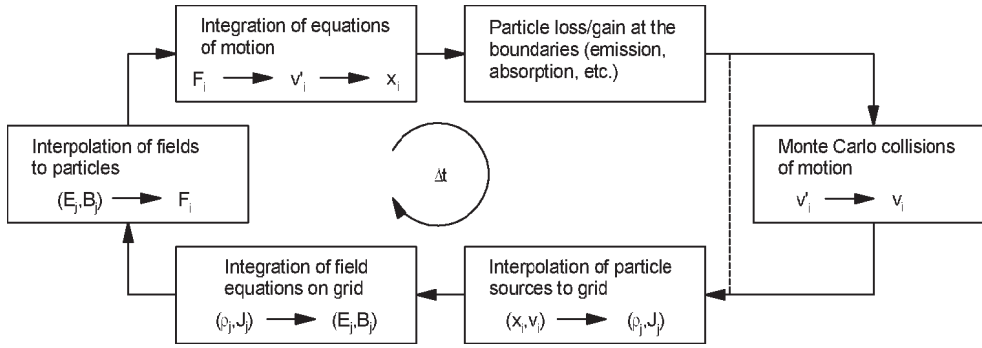


Figure 1. Flow schematic for the PIC scheme.

C K Birdsall, T Crystal, S Kuhn and W Lawson resulted in a more rigorous treatment of boundaries, which included external circuits with real impedances [16], allowing modelling of real plasma devices. At the end of this decade, the method had been refined to make the circuit plus plasma second-order accurate and self-consistent [32]. Charged-particle-neutral collisions were pioneered with simple cross sections by Boswell [3]. Commercial PIC codes were now available for beam-microwave work, and the one-dimensional Berkeley code suite based on [32] included a configurable real-time graphical user interface to facilitate computational plasma education, the descendents of which currently enjoy a user base of over 1000 worldwide.

In the 1990s, collisions were refined to include more realistic differential cross sections by Vahedi and Surendra [29]. The one-dimensional coupled circuit model was extended to two dimensions [28]. Application of object-oriented methods were applied to PIC codes to decrease the enhancement and maintenance burden while improving the fidelity of the modelling paradigm [33], and the resulting code was distributed as an open software addition to the Berkeley PIC code suite. The massively parallel computing model rose to prominence during this period as well [14, 19, 21]. By this time, two- and three-dimensional simulations were using $\sim 10^6$ – 10^8 particles on serial computers, and $\sim 10^8$ – 10^{10} particles on massively parallel platforms.

2. PIC fundamentals

In this section, the fundamentals of PIC are briefly described; the details can be found in the literature. The general flow of the PIC scheme is shown schematically in figure 1. In the PIC scheme, particles are defined in continuum space in both position and velocity. Fields are defined at discrete locations in space. However, both fields and particles are defined at discrete times. Particle and field values are advanced sequentially in time, starting from initial conditions, with the temporal scheme shown in figure 2. Particle positions and velocities are offset in time by $\Delta t/2$, which allows for the leapfrog centre different integration of the equations of motion described in equations (5) and (6). The particle equations of motion are advanced one time step, using fields interpolated from the discrete grid to the continuous particle locations. Next, particle boundary conditions such as absorption and emission are applied. If the model is collisional, the Monte Carlo collision (MCC) scheme is applied. Source terms, ρ and J , for the field equations are accumulated from the continuous particle locations to the discrete mesh locations. The fields are then advanced one time step, and the time step loop repeats.

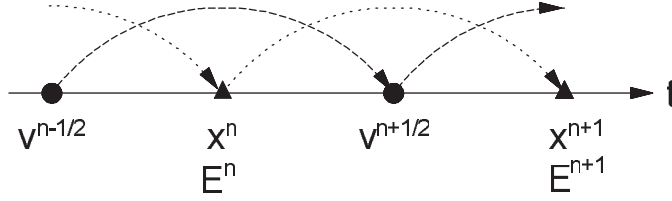


Figure 2. Schematic of the leapfrog scheme. Not shown are charge density ρ and potential Φ , which are defined at the same temporal locations as position \mathbf{x} of the particles. Also not shown are current density \mathbf{J} and magnetic field \mathbf{B} , which are defined at the temporal locations of particle velocity \mathbf{v} .

The particle positions and velocities obey the Newton–Lorentz equations of motion:

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

and

$$\frac{d}{dt} \mathbf{x} = \mathbf{v}, \quad (2)$$

where the relativistic factor is given by

$$\gamma = \sqrt{\frac{1}{1 - (v/c)^2}} = \sqrt{1 + \left(\frac{u}{c}\right)^2}, \quad (3)$$

$$u = \gamma v. \quad (4)$$

The Newton–Lorentz equations are often discretized using the second-order accurate centre difference scheme, often referred to as the *leapfrog* scheme, which has the additional merits of requiring few operations and minimal storage, since the update can be done in place.

In finite difference form, the leapfrog method is written as

$$\frac{\mathbf{u}^{t+\Delta t/2} - \mathbf{u}^{t-\Delta t/2}}{\Delta t} = \frac{q}{m} \left(\mathbf{E}^t + \frac{\mathbf{u}^{t+\Delta t/2} + \mathbf{u}^{t-\Delta t/2}}{2\gamma^t} \times \mathbf{B}^t \right), \quad (5)$$

$$\frac{\mathbf{x}^{t+\Delta t} - \mathbf{x}^t}{\Delta t} = \frac{\mathbf{u}^{t+\Delta t/2}}{\gamma^{t+\Delta t/2}} \quad (6)$$

with $\gamma^t = (\gamma^{t-\Delta t/2} + \gamma^{t+\Delta t/2})/2$. An efficient integration of the scheme, which avoids the full cross product calculation is due to Boris [2]:

$$\mathbf{u}^- = \mathbf{u}^{t-\Delta t/2} + \frac{q \Delta t \mathbf{E}^t}{2m}, \quad (7)$$

$$\mathbf{u}' = \mathbf{u}^- + \mathbf{u}^- \times \mathbf{t}^t, \quad (8)$$

$$\mathbf{u}^+ = \mathbf{u}^- + \mathbf{u}' \times \frac{2\mathbf{t}^t}{1 + \mathbf{t}^t \cdot \mathbf{t}^t}, \quad (9)$$

$$\mathbf{u}^{t+\Delta t/2} = \mathbf{u}^+ + \frac{q \Delta t \mathbf{E}^t}{2m} \quad (10)$$

with

$$\mathbf{t}^t = \hat{\mathbf{B}} \tan \left(\frac{q \Delta t}{2\gamma^t m} B^t \right). \quad (11)$$

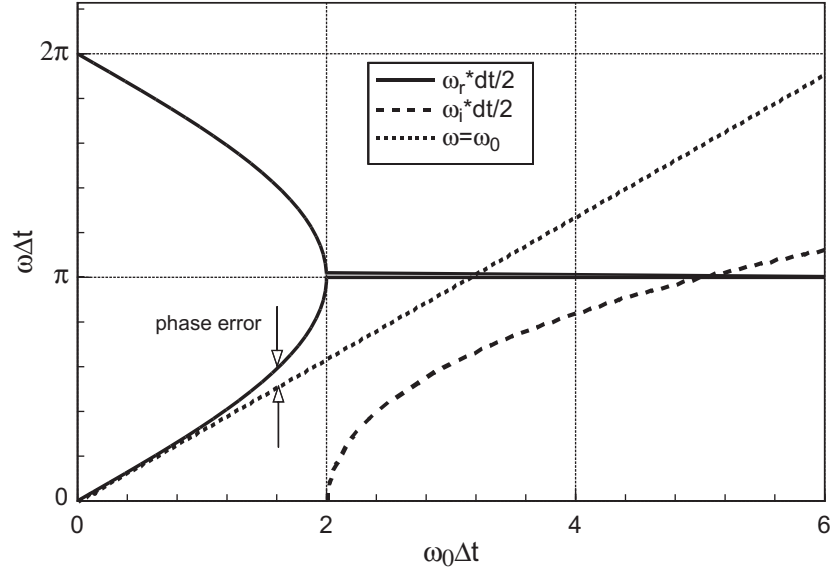


Figure 3. Angular frequency ω_r and numerical growth rate ω_i for the leapfrog scheme. Phase error is the difference between the numerical and exact frequency ω_0 .

Note that equations (8) and (9) represent a rotation only, i.e. $\mathbf{u}^- \cdot \mathbf{u}^- = \mathbf{u}^+ \cdot \mathbf{u}^+$, and only equations (7) and (10) change the magnitude of the velocity, hence the magnetic field cannot heat the particles in this scheme.

The stability of the leapfrog scheme can be shown for particles in simple harmonic motion:

$$\frac{d^2x}{dt^2} = -\omega_0^2 x. \quad (12)$$

Finite differencing using a centre difference (leapfrog) gives

$$\frac{x^{t+\Delta t} - 2x^t + x^{t-\Delta t}}{\Delta t^2} = -\omega_0^2 x^t. \quad (13)$$

Solutions are of the form

$$x^t = C \exp(-i\omega t), \quad (14)$$

$$x^{t+\Delta t} = C \exp(-i\omega(t + \Delta t)). \quad (15)$$

Using Euler's identity, the finite difference becomes

$$\sin\left(\frac{\omega\Delta t}{2}\right) = \pm \frac{\omega_0\Delta t}{2}, \quad (16)$$

so ω has an imaginary component for $\omega_0\Delta t > 2$, indicating numerical instability. Expanding equation (16) for $\omega\Delta t \ll 1$,

$$\frac{\omega\Delta t}{2} \left[1 - \frac{1}{6} \left(\frac{\omega\Delta t}{2} \right)^2 + O\left(\frac{\omega\Delta t}{2} \right)^4 \right] = \frac{\omega_0\Delta t}{2}, \quad (17)$$

we find that the phase error is quadratic for small $\omega\Delta t$, and also that a rapidly growing instability exists for $\omega_0\Delta t > 2$, as shown in figure 3. Nevertheless, self-heating was observed to cause an instability at long timescales.

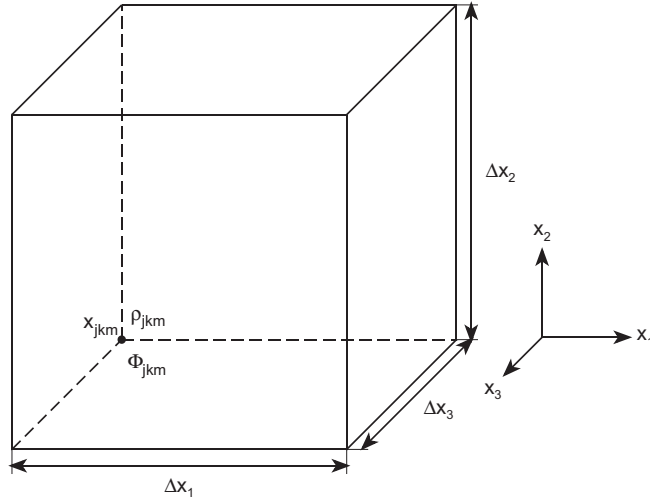


Figure 4. Computational mesh for the electrostatic model.

For electrostatic models, the mesh is defined as shown in figure 4. The source terms and potentials are known at grid nodes, with the electric fields often defined on the same nodes or along cell edges. The potential is related to the charge density by Poisson's equation, obtained by combining Gauss's law, $\nabla \cdot \epsilon \mathbf{E} = \rho$, with the definition of a scalar potential, $\mathbf{E} = -\nabla \Phi$:

$$\nabla \cdot \epsilon \nabla \Phi(\mathbf{x}, t) = \rho(\mathbf{x}, t). \quad (18)$$

Equation (18) can be solved in finite difference form in a number of ways. Using a centre difference in a one-dimensional linear homogeneous isotropic medium, we obtain

$$\frac{\Phi_{j+1} - 2\Phi_j + \Phi_{j-1}}{\Delta x^2} = -\frac{\rho_j}{\epsilon}. \quad (19)$$

Physically, the boundary conditions can be complicated [32]. For a system fully bounded by conductors, the charge is conserved:

$$\oint_S \epsilon \mathbf{E} \cdot d\mathbf{S} = \int_V \rho dV + \oint_S (\sigma_0 + \sigma_J) dS \equiv 0, \quad (20)$$

where S and V are the surface enclosing the system and the volume, respectively, and the subscripts 0 and J refer to the left and right boundary of a system with spatial indices $0 \leq j \leq J$. In general, the electric field is assumed to be zero within the ideally conducting electrodes, so that just inside the system the surface charge at each electrode is related to the surface field by

$$E_0 = \frac{\sigma_0}{\epsilon}, \quad (21)$$

$$E_J = -\frac{\sigma_J}{\epsilon}. \quad (22)$$

Mathematically, a Dirichlet boundary condition is required for a unique solution; often, one chooses a reference potential of zero at one of the electrodes (e.g. $\Phi_J = 0$).

For a non-uniform orthogonal Cartesian mesh in two dimensions, equation (18) can be centre differenced:

$$\begin{aligned} & \frac{\Phi_{j,k+1}}{\Delta x_{k+1/2} \Delta \bar{x}_k} - \frac{2\Phi_{j,k}}{\Delta x_{k+1/2} \Delta x_{k-1/2}} + \frac{\Phi_{j,k-1}}{\Delta x_{k-1/2} \Delta \bar{x}_k} + \frac{\Phi_{j+1,k}}{\Delta y_{j+1/2} \Delta \bar{y}_j} - \frac{2\Phi_{j,k}}{\Delta y_{j+1/2} \Delta x_{j-1/2}} \\ & + \frac{\Phi_{j-1,k}}{\Delta y_{j-1/2} \Delta \bar{y}_j} = -\frac{\rho_{j,k}}{\epsilon}, \end{aligned} \quad (23)$$

where

$$\Delta x_{k+1/2} = x_{k+1} - x_k \quad (24)$$

and

$$\Delta \bar{x}_k = \frac{\Delta x_{k+1/2} + \Delta x_{k-1/2}}{2} \quad (25)$$

and likewise for other subscripts and components. A Taylor expansion gives the truncation error:

$$\begin{aligned} \nabla^2 \Phi - \nabla^2 \Phi_{\text{exact}} = & \frac{\Delta x_{k+1/2} - \Delta x_{k-1/2}}{3} \Phi_{xxx} + \frac{\Delta y_{j+1/2} - \Delta y_{j-1/2}}{3} \Phi_{yyy} \\ & + \frac{(\Delta x_{k+1/2})^2 - \Delta x_{k+1/2} \Delta x_{k-1/2} + (\Delta x_{k-1/2})^2}{12} \Phi_{xxx} \\ & + \frac{(\Delta y_{j+1/2})^2 - \Delta y_{j+1/2} \Delta y_{j-1/2} + (\Delta y_{j-1/2})^2}{12} \Phi_{yyy} + \dots, \end{aligned} \quad (26)$$

which contains first-order terms for $\Delta x_{k+1/2} \neq \Delta x_{k-1/2}$. On a uniform mesh, only the second-order error terms remain, and on a slowly varying mesh, $(\Delta x_{k+1/2} - \Delta x_{k-1/2})/\Delta x_{k-1/2} \ll 1$, the first-order term may be smaller than the second-order term.

The potential can be separated into Poisson and Laplace parts [28]:

$$\Phi = \Phi_P + \sum_{\text{boundaries}} \Phi_L, \quad (27)$$

$$\nabla \cdot \epsilon \nabla \Phi_P = -\rho, \quad (28)$$

$$\nabla \cdot \epsilon \nabla \Phi_{Li} = 0. \quad (29)$$

The boundary conditions for equation (28) become $\Phi = 0$ on all boundaries. For each boundary with a Dirichlet condition, equation (29) is solved for $\Phi_i = 1$ on the equipotential surface, and $\Phi = 0$ elsewhere. The resulting Φ_{Li} can be stored and multiplied by a constant if the potential on the boundary is specified as a function of time or the result of an external circuit equation. Neumann boundary conditions are included through Φ_P . This method neglects charge induced by a driven electrode on other boundaries that are connected to an external circuit. This problem is avoided, at the expense of increased matrix size, when solving the full Poisson equation with boundaries and circuits using the method of Verboncoeur [32].

The solution to Maxwell's equations typically involves solving finite-difference time-domain (FDTD) discretizations on a Yee mesh, with the fields defined as shown in figure 5. In addition, the electric field is known at integral multiples of the time step, while the magnetic field is known at $t = (n+1/2)\Delta t$, where n is an integer. Even for an electromagnetic simulation, the initial conditions may require solution of the Poisson equation, $\nabla \cdot \epsilon \nabla \Phi = -\rho$. In addition, initial magnetic fields due to external magnets and coils may be prescribed analytically or in tabular format, and must satisfy $\nabla \cdot \mathbf{B} = 0$.

Once the initial conditions are computed, the electric and magnetic fields are then advanced in time using finite-differenced forms of Faraday's law and Ampère's law. The remaining Maxwell's equations, $\nabla \cdot \mathbf{B} = 0$ and $\nabla \cdot \mathbf{D} = \rho$ remain satisfied in time when satisfied by the initial conditions.

Starting with Ampère's law,

$$\frac{\partial \mathbf{D}}{\partial t} = \nabla \times \mathbf{H} - \mathbf{J}, \quad (30)$$

we can write a general finite-difference form relation for advancing all components of the electric flux:

$$\delta_i D_i = \delta_j H_k - \delta_k H_j, \quad (31)$$

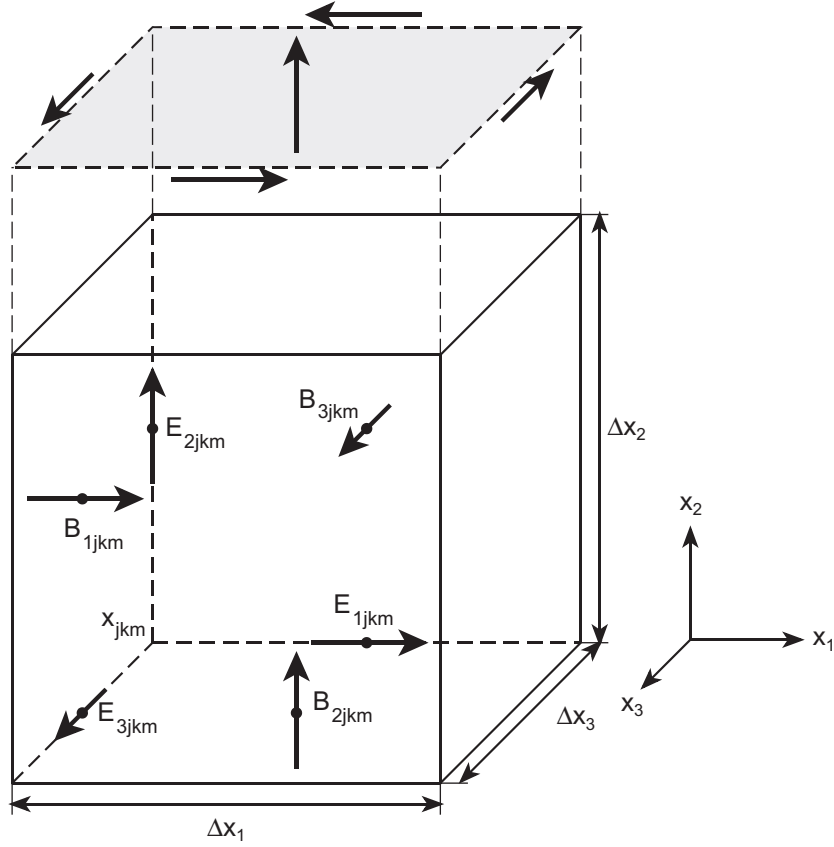


Figure 5. Fields defined on the Yee mesh. Currents, not shown, are co-located with the corresponding electric field components. Exploded view shows an integration surface.

where i , j and k denote the indices of an orthogonal right-handed set of coordinates. The full set of difference equations for Ampère's law is formed by cyclically changing these indices. In equation (31), δ_q denotes some finite-difference operator with respect to the variable q .

Faraday's law can be evaluated in the same fashion, starting from the differential form,

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}, \quad (32)$$

to form a general finite difference form of Faraday's law:

$$\delta_t B_i = -\delta_j E_k + \delta_k E_j. \quad (33)$$

The constitutive equations couple equations (31 and (33):

$$\mathbf{D} = \varepsilon \mathbf{E} \quad \text{and} \quad \mathbf{B} = \mu \mathbf{H}. \quad (34)$$

The most common implementation of equations (31) and (33) in PIC codes uses a centre difference for the differentials δ , and places the fields on the mesh as shown in figure 5, called the leapfrog algorithm [1]. That is, D , E and J are defined at the midpoints of the segments

connecting mesh nodes, while B and H are defined similarly on a mesh displaced by one half cell in each dimension. The centre difference form of Ampère's law on a uniform orthogonal mesh becomes

$$\frac{D_i^t - D_i^{t-\Delta t}}{\Delta t} = \frac{H_{k,x_j+\Delta x_j/2}^{t-\Delta t/2} - H_{k,x_j-\Delta x_j/2}^{t-\Delta t/2}}{\Delta x_j} - \frac{H_{j,x_k+\Delta x_k/2}^{t-\Delta t/2} - H_{j,x_k-\Delta x_k/2}^{t-\Delta t/2}}{\Delta x_k} - J_i^{t-\Delta t/2}. \quad (35)$$

Here, only subscripts pertaining to the direction of the derivative have been included for compactness. Similarly, the centre difference form of Faraday's law on a uniform orthogonal mesh becomes

$$\frac{B_i^{t+\Delta t/2} - B_i^{t-\Delta t/2}}{\Delta t} = -\frac{E_{k,x_j+\Delta x_j/2}^t - E_{k,x_j-\Delta x_j/2}^t}{\Delta x_j} + \frac{E_{j,x_k+\Delta x_k/2}^t - E_{j,x_k-\Delta x_k/2}^t}{\Delta x_k}. \quad (36)$$

Thus, the curl equations can be advanced in time after closing with the constitutive relations, equation (34). These equations are solved consecutively, so that the fields leapfrog forwards in time. The leapfrog algorithm is considered explicit in that the field updates only depend upon past field values. This scheme is second-order accurate in both time and space for uniform cells.

The stability of the centre difference for the wave equation describes its accuracy and stability properties. Consider the wave equation in one dimension:

$$\frac{\partial^2 \psi}{\partial t^2} = c^2 \frac{\partial^2 \psi}{\partial x^2}. \quad (37)$$

Centre differencing,

$$\psi_j^{t+\Delta t} = \left(c \frac{\Delta t}{\Delta x}\right)^2 (\psi_{j+1}^t - 2\psi_j^t + \psi_{j-1}^t) + 2\psi_j^t - \psi_j^{t-\Delta t}. \quad (38)$$

For sinusoidal waves,

$$\psi(x, t) = \exp[i(\omega t - kx)], \quad (39)$$

$$\psi_j^t = \exp[i(\omega t - \tilde{k} j \Delta x)] \quad (40)$$

with \tilde{k} the numerical wavenumber.

Combining equations (38) and (40):

$$\exp(i\omega \Delta t) = \left(c \frac{\Delta t}{\Delta x}\right)^2 [\exp(-i\tilde{k} \Delta x) - 2 + \exp(i\tilde{k} \Delta x)] + 2 - \exp(-i\omega \Delta t), \quad (41)$$

$$\frac{\exp(i\omega \Delta t) + \exp(-i\omega \Delta t)}{2} = \left(c \frac{\Delta t}{\Delta x}\right)^2 \left(\frac{\exp(i\tilde{k} \Delta x) + \exp(-i\tilde{k} \Delta x)}{2} - 1\right) + 1. \quad (42)$$

Then, the dispersion equation becomes:

$$\cos(\omega \Delta t) = \left(c \frac{\Delta t}{\Delta x}\right)^2 [\cos(\tilde{k} \Delta x) - 1] + 1, \quad (43)$$

shown in figure 6. For $c\Delta t/\Delta x = 1$, the numerical wavenumber is $\tilde{k} = \pm\omega/c = k$. Similarly, for $\Delta t \rightarrow 0$ and $\Delta x \rightarrow 0$, $\tilde{k} = \pm\omega/c = k$. For $c\Delta t/\Delta x = 0$, the dispersion approaches the limit

$$\frac{\omega \Delta x}{c} = \sqrt{2} \sqrt{1 - \cos(\tilde{k} \Delta x)}. \quad (44)$$

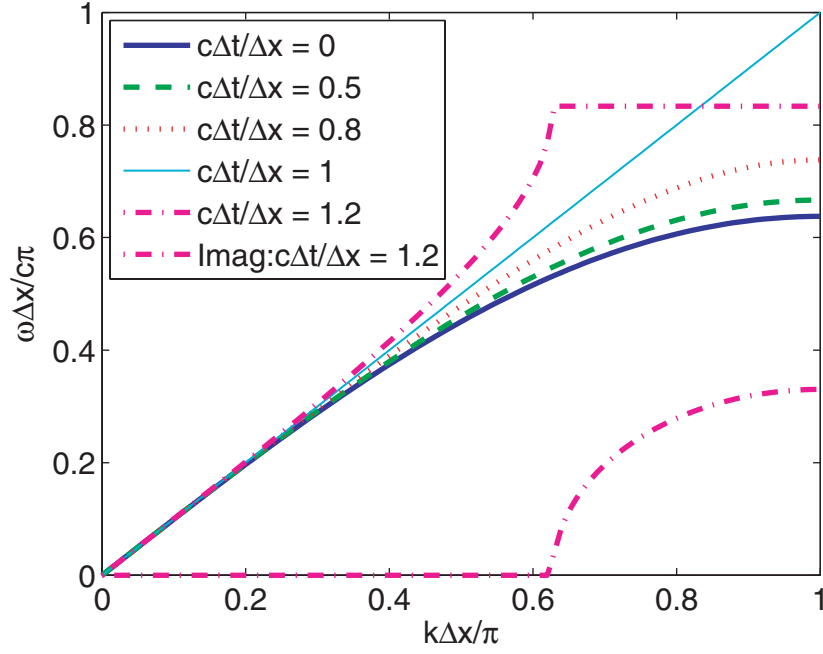


Figure 6. Vacuum dispersion curve for leapfrog difference scheme for wave equation. For $c\Delta t/\Delta x = 1.2$, both the real and imaginary parts are shown.

For $c\Delta t/\Delta x > 1$ (the Courant–Levy stability condition), a rapidly growing imaginary root is given by

$$\frac{\omega_i \Delta x}{c} = \frac{\Delta x}{c\Delta t} \cosh^{-1} \left[\left(c \frac{\Delta t}{\Delta x} \right)^2 [\cos(\tilde{k}\Delta x) - 1] + 1 \right], \quad (45)$$

which occurs for

$$\tilde{k}\Delta x > \cos^{-1} \left[1 - 2 \left(\frac{\Delta x}{c\Delta t} \right)^2 \right]. \quad (46)$$

This instability is shown with the real and imaginary parts in figure 6.

In multiple dimensions, the Courant–Levy stability criterion on the time step is given by

$$\Delta t \leq \frac{1}{c} \left(\sum_i \frac{1}{(\Delta x_i)^2} \right)^{-1/2}, \quad (47)$$

where the index i sums over the coordinate indices and Δx_i denotes the grid spacing in the i th coordinate direction.

The interaction of continuum particles with the discrete fields occurs through interpolation schemes. These interpolation methods include nearest grid point, linear, quadratic and so on, in order of increasing complexity, and the resulting particle shapes are shown in figure 7.

A linear weighting scheme for a PIC j, k is shown graphically in figure 8. Defining

$$\mathbf{w} = \mathbf{x}_i - \mathbf{X}_{jkm}, \quad (48)$$

where \mathbf{x}_i refers to the position of the i th particle, and \mathbf{X}_{jkm} is the position of the nearest lower mesh node, we can write the linear charge density weighting algorithm resulting from a single

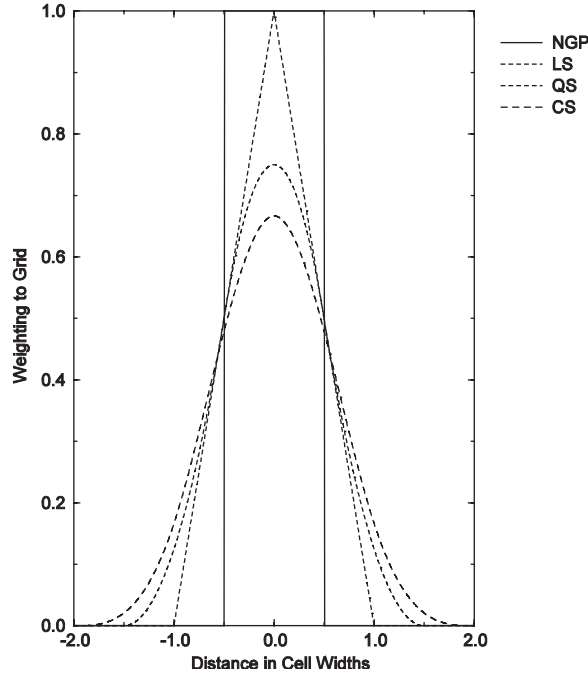


Figure 7. Particle shapes for a number of interpolation schemes: NGP is the nearest grid point, LS the linear spline, QS the quadratic spline and CS is the cubic spline.

particle being weighted to the surrounding nodes:

$$Q_{j,k,m} = q_i(1 - w_j)(1 - w_k)(1 - w_m), \quad (49)$$

$$Q_{j+1,k,m} = q_i w_j(1 - w_k)(1 - w_m), \quad (50)$$

$$Q_{j,k+1,m} = q_i(1 - w_j)w_k(1 - w_m), \quad (51)$$

$$Q_{j,k,m+1} = q_i(1 - w_j)(1 - w_k)w_m, \quad (52)$$

$$Q_{j+1,k+1,m} = q_i w_j w_k(1 - w_m), \quad (53)$$

$$Q_{j+1,k,m+1} = q_i w_j(1 - w_k)w_m, \quad (54)$$

$$Q_{j,k+1,m+1} = q_i(1 - w_j)w_k w_m, \quad (55)$$

$$Q_{j+1,k+1,m+1} = q_i w_j w_k w_m. \quad (56)$$

The charge is accumulated in this manner for all particles. The charge density is computed using $\rho_{j,k,m} = Q_{j,k,m} / V_{j,k,m}$, where $V_{j,k,m}$ is the volume of the cell centred on the j, k, m th mesh node, in the classical PIC scheme. The particle shapes of figure 7 are the charge observed by a stationary observer at a grid node for a particle drifting by at constant velocity, using the appropriate weighting scheme.

For electromagnetic models, the current is needed for Maxwell's equations. Current can be weighted using either a charge conserving method, or an algorithm equivalent to the charge density weighting algorithm. When the latter scheme is employed, charge is not identically conserved on the mesh, so periodic explicit enforcement of Gauss's law is required to reduce the buildup of dipoles.

Using the definition of w given in equation (48), and also defining

$$\Delta w = w^{t+\Delta t} - w^t \quad (57)$$

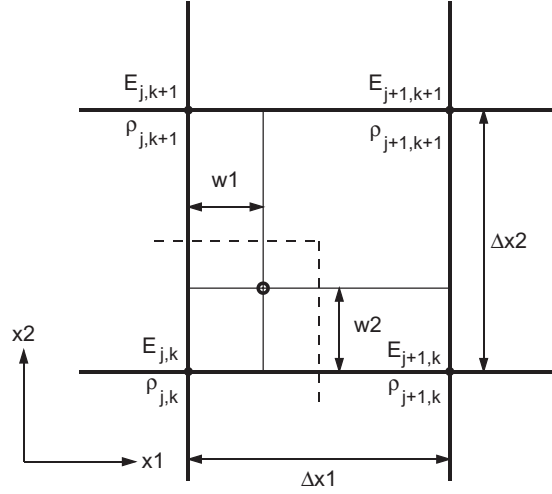


Figure 8. Weighting charge to the grid, and fields to particles.

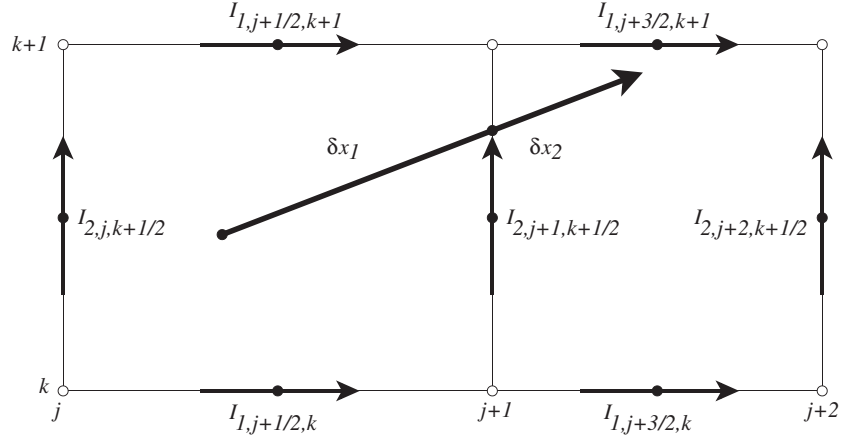


Figure 9. Current elements for a multi-cell particle motion.

and

$$\bar{w} = \frac{w^{t+\Delta t} + w^t}{2}, \quad (58)$$

we can write the two-dimensional charge conserving currents generated in the first cell due to the particle motion shown in figure 9:

$$I_{1,x_j+\Delta x_j/2,x_k} = \sum_i \frac{q_i}{\Delta t} \Delta w_1 (1 - \bar{w}_2), \quad (59)$$

$$I_{1,x_j+\Delta x_j/2,x_k+\Delta x_k} = \sum_i \frac{q_i}{\Delta t} \Delta w_1 \bar{w}_2, \quad (60)$$

$$I_{2,x_j,x_k+\Delta x_k/2} = \sum_i \frac{q_i}{\Delta t} (1 - \bar{w}_1) \Delta w_2, \quad (61)$$

$$I_{2,x_j+\Delta x_j,x_k+\Delta x_k/2} = \sum_i \frac{q_i}{\Delta t} \bar{w}_1 \Delta w_2. \quad (62)$$

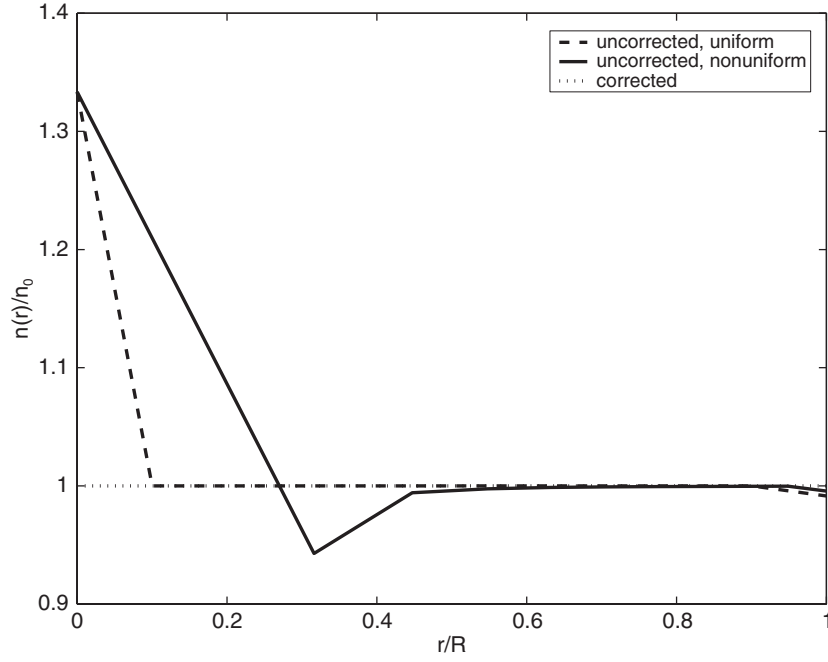


Figure 10. Particle density linearly weighted in cylindrical coordinates; the particle positions are chosen from a uniform distribution.

When cell edges are crossed, the above set of equations is applied for each cell traversed, using the segments of movement falling within the respective cell. This method is equivalent to that of Morse and Nielson [23] for particle motion within a single cell; for multiple cells it is equivalent to the method of Eastwood [10] and Villasenor and Buneman [34].

3. Recent advances

What follows is a limited subset of recent advances in particle simulation in plasmas.

3.1. Generalized weighting scheme

The classical particle weighting scheme described above uses a fixed volume for computing the charge or current density, which results in systematic errors in charge density for meshes in any grid with non-uniform volumes, including curvilinear coordinates and variable spaced meshing. The error in density for linear weighting in cylindrical coordinates is shown for uniform and non-uniform meshes in figure 10. This is a systematic error, which does *not* fluctuate. Furthermore, the magnitude of the error is independent of the mesh spacing. The error was historically corrected by a multiplier that was unique for each mesh and weighting scheme.

A general weighting scheme [31] provides a simple scheme for computing the exact density in the limit $N/J \gg 1$, where N and J are the number of particles and cells, respectively. The general scheme weights differential volume elements to the nodes using the same interpolation scheme used for the particle charge:

$$n_j = \frac{\int_r f(r) W_j(\mathbf{r}) d\mathbf{r}}{\int_r W_j(\mathbf{r}) dV}, \quad (63)$$

where $W_j(\mathbf{r})$ is an interpolation function which weights particles at position \mathbf{r} to mesh \mathbf{j} , and dV is a volume element, given in one dimension by $dV = 2\pi r dr dz$ in cylindrical coordinates and $dV = 4\pi r^2 dr$ in spherical coordinates. Here, $f(r)$ is an arbitrary continuum particle distribution, related to density in cylindrical coordinates by $n(r) = f(r)/2\pi r$.

For linear weighting in cylindrical coordinates, the density can then be written as:

$$n_j = \frac{\int_{r_{j-1}}^{r_j} f(r)((r - r_{j-1})/(r_j - r_{j-1}))dr + \int_{r_j}^{r_{j+1}} f(r)((r_{j+1} - r)/(r_{j+1} - r_j))dr}{\int_{r_{j-1}}^{r_j} 2\pi r((r - r_{j-1})/(r_j - r_{j-1}))dr + \int_{r_j}^{r_{j+1}} 2\pi r((r_{j+1} - r)/(r_{j+1} - r_j))dr}. \quad (64)$$

The edge densities are found simply by dropping the out of bounds integrals:

$$n_0 = \frac{\int_{r_0}^{r_1} f(r)((r_1 - r)/(r_1 - r_0))dr}{\int_{r_0}^{r_1} 2\pi r((r_1 - r)/(r_1 - r_0))dr} \quad (65)$$

and

$$n_N = \frac{\int_{r_{N-1}}^{r_N} f(r)((r - r_{N-1})/(r_N - r_{N-1}))dr}{\int_{r_{N-1}}^{r_N} 2\pi r((r - r_{N-1})/(r_N - r_{N-1}))dr}. \quad (66)$$

For the uniform density, $f(r) = 2\pi r$, we obtain the exact solution for all $0 \leq j \leq N$, $n_j = 1$. For more general particle distributions (specifically when f is not a linear function of r), equations (64)–(66) result in an error term that depends upon the weighting scheme; for example, the linear weighting function results in an error proportional to Δr^2 . The density obtained by the general scheme is shown in figure 10.

A similar result can be obtained for spherical coordinates, or any arbitrary mesh.

3.2. Collisions: Monte Carlo collision scheme

Charged-particle–neutral collisions were first integrated into the PIC scheme by Boswell and Morey [3] using step probabilities, and later refined by Vahedi and Surendra [29] with energy dependent cross sections. The collision probability of the i th particle can be written as

$$P_i = 1 - \exp[-n_g(\mathbf{x})\sigma_T(\mathcal{E}_i)v_i\Delta t], \quad (67)$$

where the total cross section is the sum over all processes, $\sigma_T(\mathcal{E}_i) = \sum_j \sigma_j(\mathcal{E}_i)$. Here, $n_g(\mathbf{x})$ is the spatially varying target density, v_i the incident speed, \mathcal{E}_i the kinetic energy, and Δt is the time interval.

There is a finite probability that the i th particle will undergo more than one collision in the time interval Δt . For nearly lossless collisions, such as elastic scattering with a massive target, the probability of n collisions in a time interval Δt is P_i^n , so the total number of missed collisions can be written as

$$r_i \approx \sum_{k=2}^{\infty} P_i^k = \frac{P_i^2}{1 - P_i}. \quad (68)$$

If $P_i \ll 1$, $r_i \approx P_i^2$ provides a measure of the under-representation of the collision operator. Hence, traditional PIC-MCC codes are constrained by $\nu_{T,\max}\Delta t \ll 1$ for accuracy, where $\nu_{T,\max} = \max[n_g(\mathbf{x})\sigma_T(\mathcal{E}_i)v_i]$ is the maximum collision frequency in space and energy.

Computing the collision probability for each particle for each time step is computationally expensive, since it involves computing the particle energy, a square root to obtain the speed, and either interpolation of tabled cross sections or computation of a curve fit for each process for every particle. The cost of the MCC can exceed significantly the cost of integrating the equations of motion, so we seek a more efficient method.

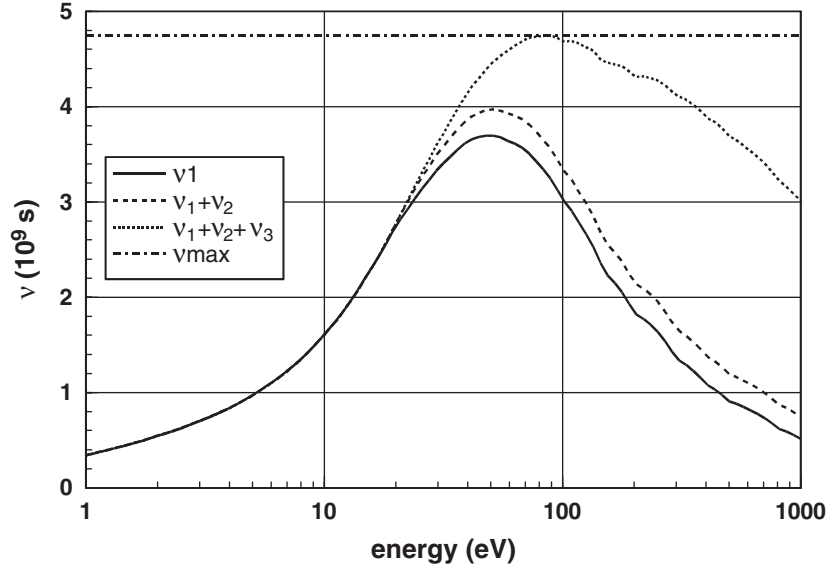


Figure 11. Summed collision frequencies for the null collision method.

Defining a maximum collision frequency in space and energy,

$$\nu_{\max} = \max_{\mathbf{x}}(n_g(\mathbf{x})) \max_{\mathcal{E}}(\sigma_T(\mathcal{E})\nu), \quad (69)$$

we can write a total collision probability independent of particle energy and position, $P_T = 1 - \exp(-\nu_{\max}\Delta t)$. The $N_c = P_T N$ particles can be chosen at random from the particle list, and the collision dynamics calculations can be performed only on the subset. Once the particles undergoing collisions have been selected, the type of collision for each particle is determined by choosing a random number, $0 \leq R \leq \nu_{\max}$. R is mapped onto the collision frequencies shown in figure 11. This method is called the *null collision method*, because an energy dependent null collision frequency has been added in order to form the energy independent probability, which eliminates the computation of energies and cross sections for all particles at each time step.

Next, we consider the collision dynamics of electron–neutral collisions. A number of electron–neutral collision events are possible, including elastic scattering ($e + A \rightarrow e + A$), excitation ($e + A \rightarrow e + A^*$) and ionization ($e + A \rightarrow e + A^+ + e$). Here, e represents an electron, A represents a neutral atom, A^* is an excited state of A and A^+ is the singly ionized state of A .

First, consider electron–neutral elastic collisions. The differential cross section is required to compute the final velocity of the incident electron. For elastic scattering of electrons in argon, one empirical cross section is [26]

$$\frac{\sigma(\mathcal{E}_i, \chi)}{\sigma(\mathcal{E}_i)} = \frac{\mathcal{E}_i}{4\pi[1 + \mathcal{E}_i \sin^2(\chi/2)] \ln(1 + \mathcal{E}_i)} \quad (70)$$

with incident electron energy \mathcal{E}_i in eV and χ being the scattering angle. It should be noted that many other choices are possible; for example, see [12] for differential ionization cross sections in a number of gases. The cumulative distribution function is

$$R = \frac{\int_0^\chi \sigma(\mathcal{E}, \chi) \sin \chi \, d\chi}{\int_0^\pi \sigma(\mathcal{E}, \chi) \sin \chi \, d\chi}. \quad (71)$$

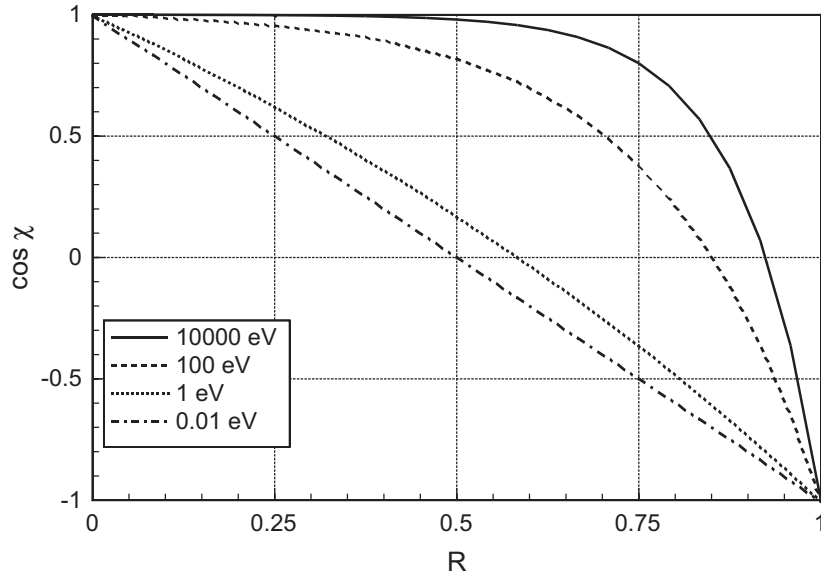


Figure 12. Scattering angle distribution for a range of energies.

If R is a uniformly distributed random number $0 \leq R \leq 1$, the scattering angle becomes

$$\cos \chi = \frac{2 + \mathcal{E}_i - 2(1 + \mathcal{E}_i)^R}{\mathcal{E}_i}. \quad (72)$$

The scattering angles for electrons incident on argon neutrals are shown in figure 12. The angular distribution varies from isotropic at 10 mV to small-angle dominated at 10 kV. The azimuthal angle is uniformly distributed, $0 \leq \theta \leq 2\pi$.

Once the scattering angle is specified, the fractional energy loss in the scattering event can be computed by classical collision mechanics [22]:

$$\Delta \mathcal{E} = \frac{2m}{M} (1 - \cos \chi) \mathcal{E}_i. \quad (73)$$

Next, we consider electron–neutral inelastic collisions, such as excitation and ionization. For ionization, the energy balance is $\mathcal{E}_f = \mathcal{E}_i - \mathcal{E}_2 + \mathcal{E}_N - \mathcal{E}_+ - \mathcal{E}_{iz}$, where \mathcal{E}_i and \mathcal{E}_f are the initial and final primary electron energies, \mathcal{E}_2 the energy of the ionization electron, \mathcal{E}_N the neutral energy, \mathcal{E}_+ the ion energy and \mathcal{E}_{iz} is the ionization threshold. Since the mass of the electron is small compared to the mass of the neutral, $m \ll M$, we neglect the momentum change of the neutral, so that $\mathcal{E}_N = \mathcal{E}_+$. For excitation, the energy balance becomes $\mathcal{E}_f = \mathcal{E}_i - \mathcal{E}_{ex}$, where \mathcal{E}_{ex} is the excitation threshold.

For ionization at low \mathcal{E}_i , a differential cross section may be chosen, which is of the form [25]

$$S(\mathcal{E}_i, \mathcal{E}_2) = \frac{\sigma_{iz}(\mathcal{E}_i) B(\mathcal{E}_i)}{\arctan \{ [\mathcal{E}_i - \mathcal{E}_{iz}] / [2B(\mathcal{E}_i)] \} [\mathcal{E}_2^2 + B^2(\mathcal{E}_i)]}. \quad (74)$$

Here, $B(\mathcal{E}_i)$ is a known function for many gases. Similar differential ionization cross sections for a series of gases are given in [12]. Inverting the distribution,

$$R = \frac{\int_0^{\mathcal{E}_2} S(\mathcal{E}_i, \mathcal{E}'_2) d\mathcal{E}'_2}{\int_0^{(\mathcal{E}_i - \mathcal{E}_{iz})/2} S(\mathcal{E}_i, \mathcal{E}'_2) d\mathcal{E}'_2} \quad (75)$$

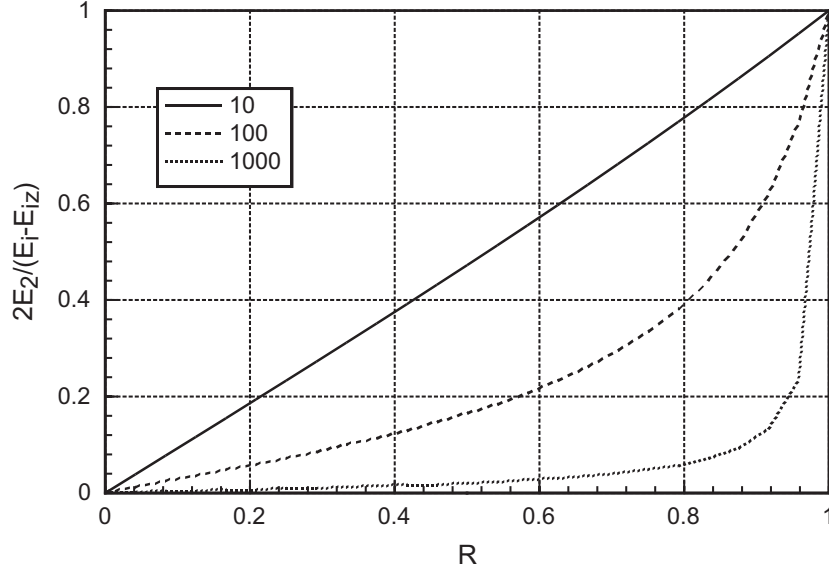


Figure 13. Normalized scattered electron energy distribution for $10 \leq \mathcal{E}_i \leq 1000$ eV.

gives the energy of the ionization electron:

$$\mathcal{E}_2 = B(\mathcal{E}_i) \tan \left[R \arctan \left(\frac{\mathcal{E}_i - \mathcal{E}_{iz}}{2B(\mathcal{E}_i)} \right) \right]. \quad (76)$$

Although the primary and ionization electrons are indistinguishable, we have chosen the convention that the more energetic electron is the primary. In figure 13, the energy of the ionization electron normalized to the maximum energy, $\mathcal{E}_{2,\max} = \frac{1}{2}(\mathcal{E}_i - \mathcal{E}_{iz})$, is plotted for a number of incident electron energies. The energy is distributed uniformly at low energies, and the normalized ionization electron energy is reduced significantly at higher energies.

Ion–neutral collisions are similar to electron–neutral collisions, except the collision mechanics must be performed in the rest frame of the neutral since the momenta are similar, $v_i \sim v_N$. Examples of ion–neutral collisions include elastic scattering ($A^+ + A \rightarrow A^+ + A$), and charge exchange ($A^+ + A \rightarrow A + A^+$). Here, A represents the neutral atom and A^+ represents the singly ionized state of A . Many other reactions are possible, but are not addressed here.

First, the velocity of the ion is converted to the rest frame of the neutral using $\mathbf{v}'_i = \mathbf{v}_i - \mathbf{v}_N$. The ion energy is then computed using $\mathcal{E}_i = M_i \mathbf{v}'_i \cdot \mathbf{v}'_i / 2$. For elastic scattering, the scattered ion energy can be written as

$$\mathcal{E}_f = \left(1 - \frac{2M_i M_N}{(M_i + M_N)^2} (1 - \cos \Theta) \right) \mathcal{E}_i \quad (77)$$

where $\cos \chi = \sqrt{1 - R}$ for isotropic scattering, and Θ is the scattering angle in the centre of mass frame. The subscripts i and N refer to the incident ion and neutral, respectively. For $M_i = M_N$, $\Theta = 2\chi$, where χ is the scattering angle in the laboratory frame, and the scattered ion energy becomes $\mathcal{E}_f = \mathcal{E}_i \cos^2 \chi$. The azimuthal angle is chosen randomly, and the scattered velocity in the rest frame of the neutral is constructed using the angle Θ and the random azimuthal angle in conjunction with the magnitude, satisfying $\mathcal{E}_f = M_i \mathbf{v}'_f \cdot \mathbf{v}'_f / 2$. The resulting velocity is then converted back to the laboratory frame using $\mathbf{v}_f = \mathbf{v}'_f + \mathbf{v}_N$.

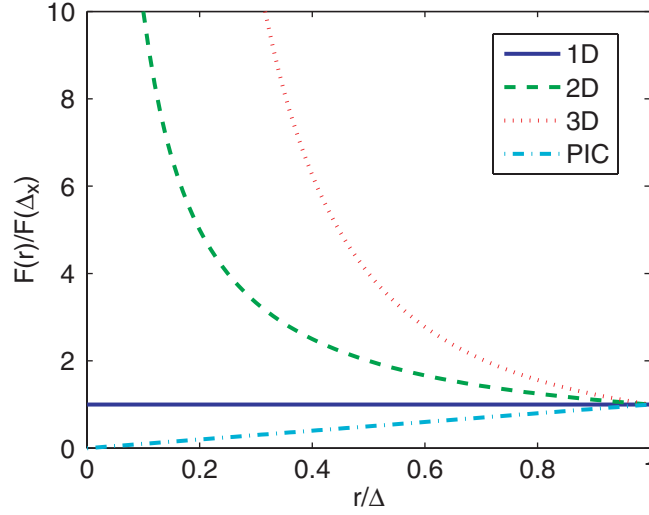


Figure 14. Force between two PIC particles separated by a distance r using linear weighting, compared to Coulomb's law in one, two and three dimensions.

Charge exchange uses the same process for computing the probability, but the neutral velocity is chosen from a prescribed analytic distribution. The neutral identity is then exchanged with the ion.

3.3. Coulomb collision models

The interparticle force within a cell for linearly weighted particles in the PIC scheme is compared to Coulomb's law in one, two and three dimensions in figure 14. The PIC scheme under-represents the force, with a force which declines linearly to zero as the particle separation approaches zero, for particles separated by less than a cell. This has the consequences of requiring resolution of the Debye length with the mesh to reduce the level of the error when Coulomb interactions are important.

This problem is avoided when the fields are computed from a direct solution of Coulomb's law. Because that is an order N^2 calculation, it is impractical compared to the order $N \log N$ PIC calculation. Other schemes have been devised to reduce the error. Takizuka and Abe [27] described a scheme to compute binary collisions within a cell, which required the time step $\Delta t \ll 1/v$, in addition to sorting particles by cells. In this method, pairs within cells are randomly matched, and each collision angle is calculated, making the method computationally intensive.

Later, Nanbu [24] devised a method for computing only large angle collisions explicitly, by summing over small angle collisions. The method allows an independent choice of time step. The number of targets encountered per time step is $N = n_t \pi b_{\max}^2 v_R \Delta t$ for target density n_t . The mean square deflection angle is

$$\langle \theta_1^2 \rangle = 2 \left(\frac{b}{b_{\max}} \right)^2 \ln \left(\frac{b_{\max}}{b_0} \right), \quad (78)$$

where $b_{\max} \approx \lambda_D$ and $b_0 = |q_1 q_2| / (2\pi \epsilon_0 m_R v_R^2)$, m_R being the reduced mass and v_R the relative speed. An isotropy parameter, $s = n_t v_R \pi b_0^2 \ln(\lambda_D/b_0)$, indicated collisions were

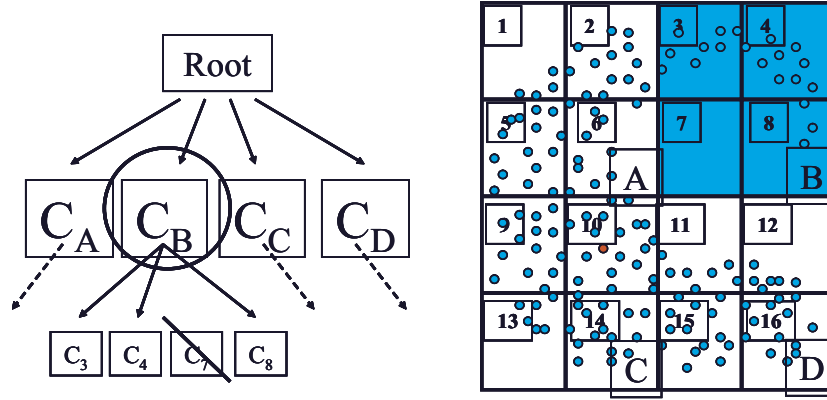


Figure 15. Treecode method uses clusters at long interaction distances and the direct Coulomb's law for short range interactions.

negligible for small s , and for large s approached an isotropic distribution. The Nanbu method requires sorting, and has $N \log N$ efficiency for N particles.

More recently, Christlieb *et al* [7] described a gridless treecode method to treat the full set of fields. The treecode model is illustrated in figure 15. In the treecode model, clusters of particles interact at a distance, while individual particles interact when in close proximity. The homogeneous solution is obtained via a boundary integral method. This allows the method to include both the short and long range Coulomb collision operators, while still resulting in an $N \log N$ method for N particles, comparable to the PIC scheme. Because the method is gridless, it offers an efficient scheme for modelling non-conformal boundaries. The method requires spatial sorting of particles. Many details have not yet been worked out for the treecode method, such as the consequence for noise properties resulting from the short range Coulomb interaction.

A comparison of the treecode scheme with the classical PIC model is shown in figure 16 for a one-dimensional virtual cathode oscillation. Note that the PIC method results in a sharper oscillation, because the short range Coulomb collisions in the treecode result in lower peak densities and, consequently, a weaker interaction at the peak. The higher peak densities at the turning point in the PIC code result in exaggeration of the current modulation, which can be seen as gaps in the velocity–position phase space. What is not shown is that a direct application of Coulomb's law agrees well with the treecode result; the PIC result converges to the treecode result for $\Delta x \rightarrow 0$.

3.4. Secondary emission

Recent advances in modelling of plasma surface interactions have included secondary emission due to electron and ion impact, as well as in sputtering and desorption models. All these models are algorithmically similar, so we will address only a general secondary model due to electron impact here [11].

The secondary emission coefficient due to electron impact can be written as

$$\delta(\mathcal{E}, \theta) = \delta_{\max 0} \left(1 + k_{s\delta} \frac{\theta^2}{2\pi} \right) W(w). \quad (79)$$

Here, the incident energy is given by \mathcal{E} , the angle with respect to the surface normal is θ , $k_{s\delta}$ is a surface smoothness parameter described below, k a curve-fit parameter also described below,

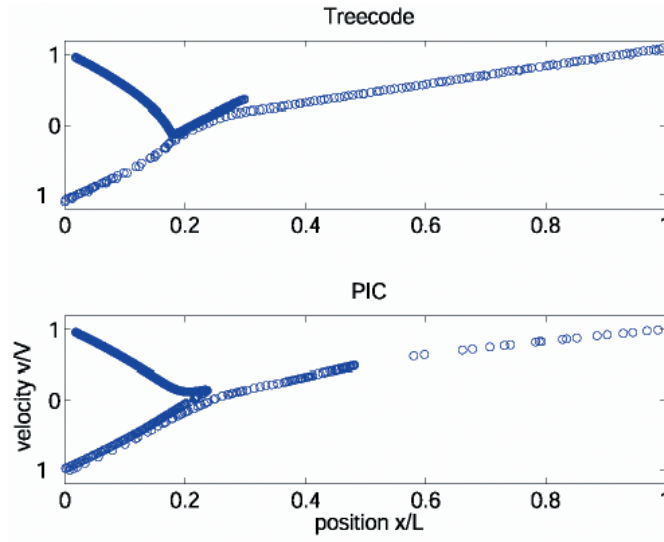


Figure 16. Comparison of treecode (top) and PIC (bottom) models for a one-dimensional virtual cathode oscillation.

and $\delta_{\max 0}$ the peak coefficient at normal incidence at the energy $\mathcal{E}_{\max 0}$. The energy dependence appears implicitly in the right-hand side of equation (79) through

$$W(w) = \begin{cases} (w \exp(1 - w))^k, & w \leq 3.6, \\ \frac{1.125}{w^{0.35}}, & w > 3.6, \end{cases} \quad (80)$$

where the normalized energy, w , is given by

$$w = \frac{\mathcal{E} - \mathcal{E}_0}{\mathcal{E}_{\max 0}(1 + k_{sw}\theta^2/2\pi) - \mathcal{E}_0}, \quad (81)$$

where \mathcal{E}_0 is the secondary emission threshold and k_{sw} is a surface-smoothness parameter similar to $k_{s\delta}$. Both $k_{s\delta}$ and k_{sw} vary between 0 for very rough surfaces and 2 for polished surfaces. Typical values are close to 1. The exponent k in equation (80) is given by

$$k = \begin{cases} 0.56, & w < 1, \\ 0.25, & 1 \leq w \leq 3.6. \end{cases} \quad (82)$$

The energy dependence is shown in figure 17, and the angular dependence is shown in figure 18 for electron generated secondaries. Similar yield relations can be obtained for ion-induced secondaries, sputtering and desorption.

A schematic of the energy distribution of emitted particles is shown in figure 19. Reflected particles typically comprise about 3% of the total particles, and are incident electrons reflected at the surface, so they have the full incident energy. Backscattered primaries, comprising about 7% of the ejected electrons, are incident electrons that have penetrated the surface and scattered several times before scattering out of the surface, and consequently they have energy between zero and the incident energy. True secondaries, comprising about 90% of the ejected population, gain sufficient energy to escape the surface by interactions between the incident particle, the free and bound electron populations, and the solid-state lattice. The distribution of the true secondaries is taken as a Maxwell–Boltzmann distribution at a temperature T_e of

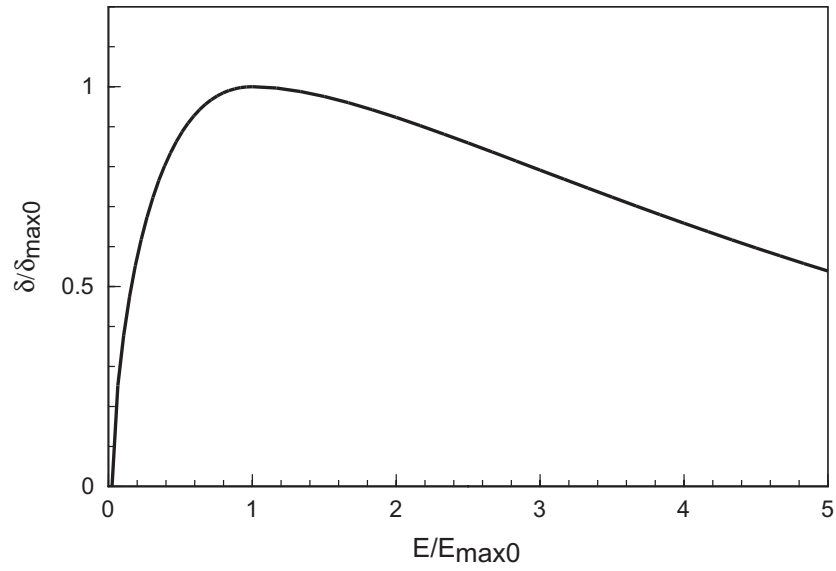


Figure 17. Energy dependence of the secondary emission coefficient.

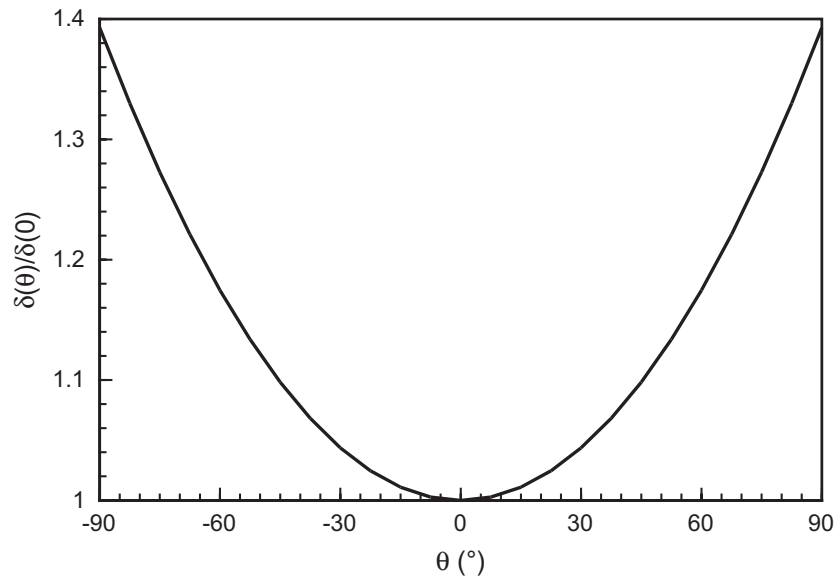


Figure 18. Angular dependence of the secondary emission coefficient.

the order of tens of eV:

$$f(\mathcal{E}) = \frac{\mathcal{E}}{(k_B T_e)^2} \exp\left(-\frac{\mathcal{E}}{k_B T_e}\right), \quad (83)$$

where k_B is the Boltzmann constant. The angular distribution is taken to be isotropic:

$$g(\theta) = \frac{1}{2} \cos(\theta) \quad (84)$$

with θ the angle from the surface normal. The azimuthal angle is chosen randomly.

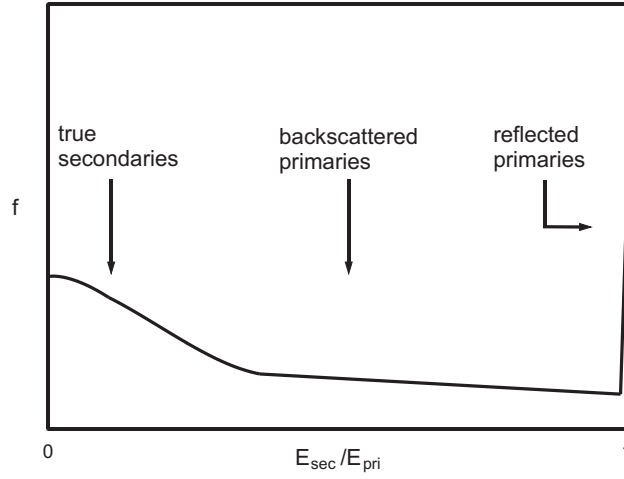


Figure 19. Schematic of the energy distribution of electrons ejected by electron-surface impact. Here, $E_{\text{pri}} = \mathcal{E}$ is the energy of the incident particle.

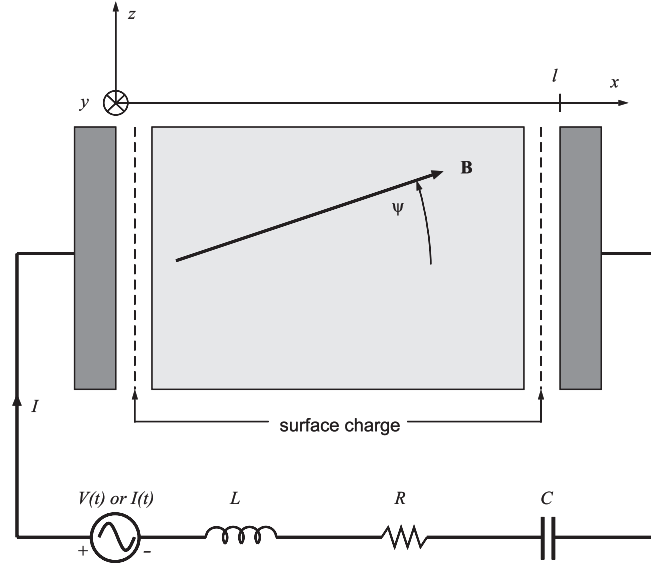


Figure 20. RLC circuit schematic for a one-dimensional model.

3.5. External circuit scheme

Although specified Dirichlet and Neumann boundary conditions for Poisson's equation are straightforward, modelling of devices often requires including resistive and reactive external impedances. An external circuit model was developed to provide this capability [32]. The model is shown schematically for one dimension in figure 20. In this scheme, the gridded potential is defined at grid nodes, Φ_j , and used in equation (19) along with the boundary conditions of equations (21) and (22). The scheme allows for the usual Dirichlet voltage using a short circuit with $V(t) = \Phi_0(t)$ specified. An ideal current source $I(t)$ is obtained by setting

$$E_{1/2}^t = \frac{\Phi_0^t - \Phi_1^t}{\Delta x} = \frac{\sigma_0^t + \rho_0^t \Delta x / 2}{\epsilon} \quad (85)$$

and

$$\sigma_0^t = \sigma_0^{t-\Delta t} + I(t) - \int_{t-\Delta t}^t J_{\text{plasma}} dt, \quad (86)$$

where σ_0^t is the surface charge on the electrode at $x = 0$ at time t , and J_{plasma} is the wall current due to absorption of charged particles.

A general RLC circuit is obtained by solving a coupled second-order ordinary differential equation for the charge Q in the circuit shown in figure 20:

$$L \frac{d^2 Q(t)}{dt^2} + R \frac{dQ(t)}{dt} + \frac{Q(t)}{C} = V(t) + \Phi_J(t) - \Phi_0(t), \quad (87)$$

where L is the inductance, R the resistance and C the capacitance in the circuit and $V(t)$ is the specified ideal voltage source. Other lumped element circuits can be written in a similar fashion. Equation (87) can be finite differenced using a second-order backward difference:

$$Q^t = \frac{V(t) + \Phi_J^t - \Phi_0^t - K^t}{\alpha_0} \quad (88)$$

with

$$K^t = \alpha_1 Q^{t-\Delta t} + \alpha_2 Q^{t-2\Delta t} + \alpha_3 Q^{t-3\Delta t} + \alpha_4 Q^{t-4\Delta t}, \quad (89)$$

$$\alpha_0 = \frac{9}{4} \frac{L}{\Delta t^2} + \frac{3}{2} \frac{R}{\Delta t} + \frac{1}{C}, \quad (90)$$

$$\alpha_1 = -6 \frac{L}{\Delta t^2} - 2 \frac{R}{\Delta t}, \quad (91)$$

$$\alpha_2 = \frac{11}{2} \frac{L}{\Delta t^2} - \frac{1}{2} \frac{R}{\Delta t}, \quad (92)$$

$$\alpha_3 = -2 \frac{L}{\Delta t^2}, \quad (93)$$

$$\alpha_4 = \frac{1}{4} \frac{L}{\Delta t^2}. \quad (94)$$

The second-order backward difference has the property of being stable over all values of numerical and real physical parameters, although for small Δt the stability is neutral, and hence can be tenuous. Many other difference schemes are also possible.

3.6. Noise control

Statistical noise, proportional to $N^{-1/2}$, remains a persistent problem in particle simulation of plasmas. Fluctuation levels in simulations exceed those in the physical system often by many orders of magnitude, and can lead to numerical effects such as artificial inflation of electron temperatures. Spatial filtering schemes have long been employed in planar models to reduce short wavelengths [1]. A Fourier space filter operates on the charge density in k -space to obtain the next generation (filtered) charge density from the previous one:

$$\rho^{n+1}(k) = \rho^n(k) S^2(k), \quad (95)$$

where the smoothing function $S(k)$ might be a function tuned to suppress short wavelengths (a_1 term) and boost long wavelengths back to $S = 1$ (a_2 term):

$$S(k) = \exp \left(a_1 \sin^2 \left(\frac{k \Delta x}{2} \right) - a_2 \tan^4 \left(\frac{k \Delta x}{2} \right) \right). \quad (96)$$

An example of a digital filter in a planar system is written as:

$$\rho_j^{n+1} = \frac{W\rho_{j+1}^n + \rho_j^n + W\rho_{j-1}^n}{1 + 2W}, \quad (97)$$

where W is used to determine the strength of the filter. For a binomial filter, $W = 0.5$ is used, leading to the common binomial digital filter:

$$\rho_j^{n+1} = \frac{\rho_{j+1}^n + 2\rho_j^n + \rho_{j-1}^n}{4}. \quad (98)$$

These digital filters can be applied in multiple passes to achieve additional smoothing; as the number of passes approaches infinity, the binomial filter results in a Gaussian shape. Values of $W < 0$ result in a compensating filter.

However, these schemes are not effective in curvilinear coordinates. A scheme for filtering short wavelengths in cylindrical coordinates was recently developed [30]. The method can be applied to general systems in the same way. For a one-dimensional system, the filtered density is written in terms of the previous density in a form that will not perturb a uniform density:

$$\begin{aligned} \rho_0^{n+1} &= (1 - \alpha_0)\rho_0^n + \alpha_0\rho_1^n, \\ \rho_j^{n+1} &= \alpha_j\rho_{j-1}^n + (1 - 2\alpha_j)\rho_j^n + \alpha_j\rho_{j+1}^n, \quad 0 < j < N, \\ \rho_N^{n+1} &= \alpha_N\rho_{N-1}^n + (1 - \alpha_N)\rho_N^n, \end{aligned} \quad (99)$$

where j is the grid index and N the number of grid cells. The coefficients, α_j , can be determined by requiring conservation of charge:

$$\begin{aligned} \alpha_0 &= \frac{1}{2}, \\ \alpha_1 &= \frac{\alpha_0 V_0}{V_1}, \\ \alpha_j &= \frac{2\alpha_{j-1}V_{j-1}}{V_j} - \frac{\alpha_{j-2}V_{j-2}}{V_j}, \quad 2 \leq j < N, \\ \alpha_N &= \frac{\alpha_{N-1}V_{N-1}}{V_N}, \end{aligned} \quad (100)$$

where V_j is the volume of the j th node corresponding to ρ_j . Here, one free parameter remained, so the coefficient $\alpha_0 = \frac{1}{2}$ was chosen so that the binomial filter is recovered in the planar limit.

The shape of a particle is little changed by a single pass of the filter, as shown in figure 21. Empirically, hundreds of passes are required to make the shape smooth, which becomes computationally expensive. The filter can be modified to apply multiple passes in a single matrix multiplication:

$$\rho^n = A^n \rho^0, \quad (101)$$

where the matrix A can be precomputed once:

$$A = \begin{pmatrix} 1 - \alpha_0 & \alpha_0 & 0 & \cdots & \cdots \\ \alpha_1 & 1 - 2\alpha_1 & \alpha_1 & & \cdots \\ 0 & & \cdots & & 0 \\ \cdots & & \alpha_{N-1} & 1 - 2\alpha_{N-1} & \alpha_{N-1} \\ & \cdots & 0 & \alpha_N & 1 - \alpha_N \end{pmatrix}. \quad (102)$$

The coefficients for a 100-pass filter for a 20 cell uniformly spaced cylindrical system are shown in figure 22. A horizontal slice (row) of the coefficient matrix shows the contribution of the unfiltered data to a particular filtered element, while a vertical slice (column) shows the redistribution of a unit of charge density at a particular radius to other radii. Note that

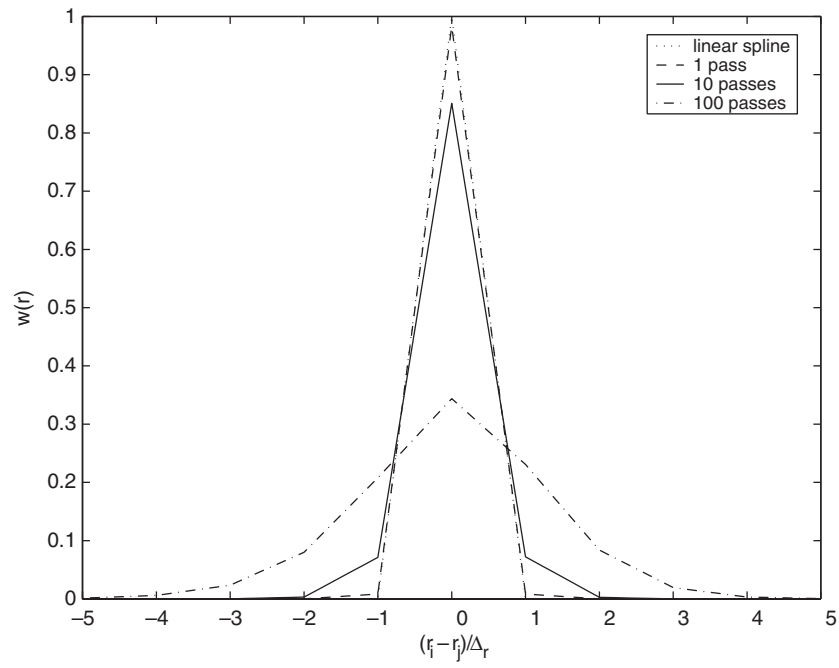


Figure 21. Shape of a standard linearly weighted particle, and shape after 1, 10 and 100 passes of the $\alpha_0 = \frac{1}{2}$ cylindrical filter. The linear spline and 1-pass curve are nearly co-located.

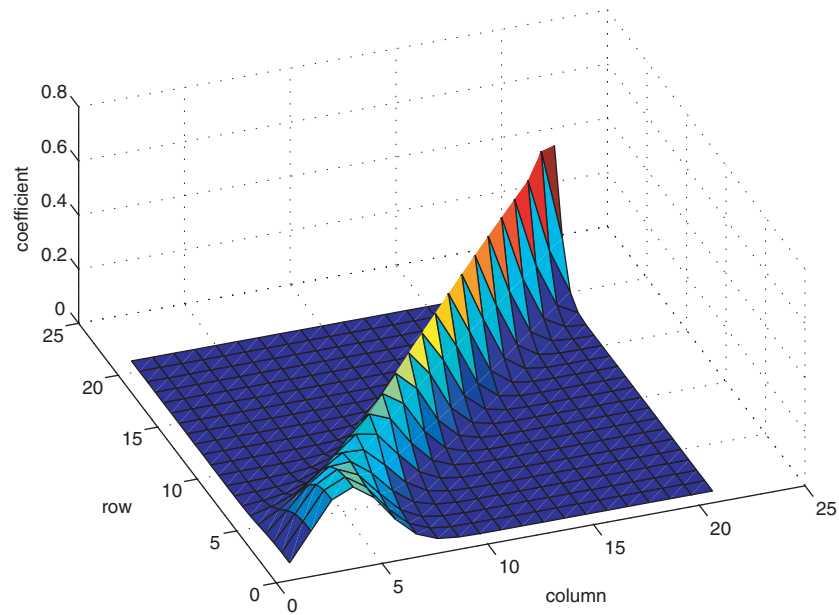


Figure 22. Multi-pass digital filter coefficients for a 100 pass filter on a 20 cell uniformly spaced radial mesh in cylindrical coordinates, with $\alpha_0 = \frac{1}{2}$.

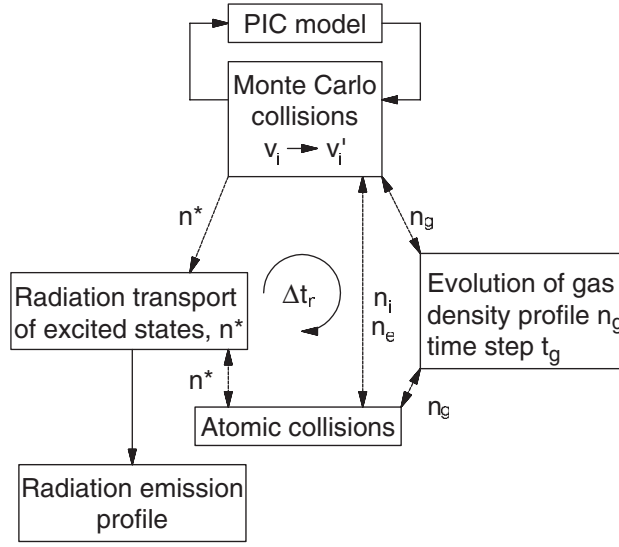


Figure 23. Flow diagram for radiation transport model.

charges are distributed over the largest distance at small radii, and the filter becomes weaker at larger radii, which is a desirable result since noise is most pronounced near the axis, being proportional to $1/r$.

3.7. Radiation transport

A radiation transport hybrid model was recently implemented with the PIC scheme to study plasma lighting devices [18], as shown in figure 23. The densities shown, n_e , n_i , n_g and n^* are the electron, ion, ground state and excited state densities, respectively. In this scheme, the ground and excited states are followed as fluid species with their own timescales. The background density is solved at intervals given by Δt_g , while the excited states are solved on the timescale Δt_r , such that $\Delta t_{\text{PIC}} < \Delta t_r < \Delta t_g$. The transport of the excited states is performed using a propagator method [15] with complete frequency redistribution. This method can include effects of Lorentz (pressure) broadening and Doppler (thermal) broadening of the excited state distributions. The excited state density can be written as

$$\frac{\partial}{\partial t} n^*(\mathbf{r}, t) + \nabla \cdot \mathbf{\Gamma}(\mathbf{r}, t) = S(\mathbf{r}, t) - \frac{n^*(\mathbf{r}, t)}{\tau_v} + \frac{1}{\tau_v} \int n^*(\mathbf{r}', t) G(\mathbf{r}, \mathbf{r}') d\mathbf{r}', \quad (103)$$

where n^* is the excited state density, $\mathbf{\Gamma}$ the excited state flux, τ_v the vacuum radiative decay frequency and G is the kernel function of radiation transport. The source term S is coupled to the Monte Carlo collisions within the PIC model, and consequently contains statistical noise. With a piecewise constant approximation in a discrete mesh $0 \leq l, j \leq N_x$ and $0 \leq m, k \leq N_y$, we obtain

$$\frac{\partial}{\partial t} n^*(\mathbf{r}, t) + \nabla \cdot \mathbf{\Gamma}(\mathbf{r}, t) = S(\mathbf{r}, t) - \sum_{l=1}^{N_x} \sum_{m=1}^{N_y} n_{lm}^*(t) \int_{lm} G(\mathbf{r}_{jk}, \mathbf{r}') d\mathbf{r}'. \quad (104)$$

A similar equation can be written for individual excited states. The integral in equation (104) can be computed infrequently as the opaque densities (usually the ground state) change in order to improve performance.

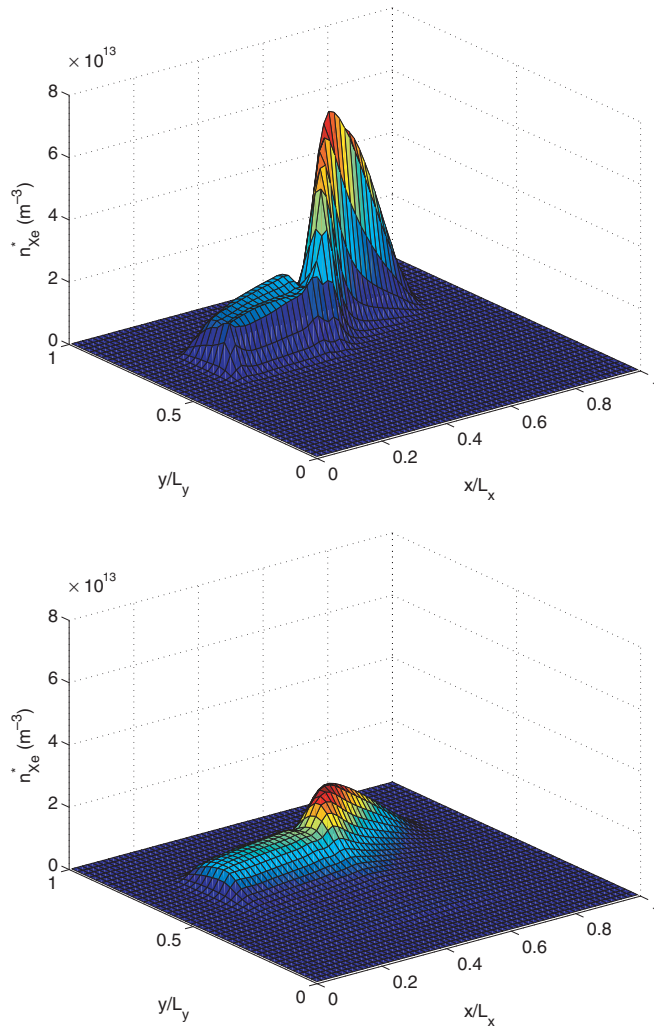


Figure 24. Comparison of xenon excited state density profile in a plasma display panel at $5.7 \mu\text{s}$ without (top) and with (bottom) radiation transport.

In figure 24, the excited state density profiles of xenon, n_{Xe}^* , are compared at $t = 5.7 \mu\text{s}$ with radiation transport off (top) and on (bottom), for a plasma panel display cell [17]. The sustain electrodes are beyond $y/L_y = 1$ and the address electrode is below $y/L_y = 0$. The gas pressure was 500 Torr, with 4% Xe in Ne, with an applied voltage of 200 V in a $5 \mu\text{s}$ pulse, gap width $L_y = 60 \mu\text{m}$ and cell width $L_x = 125 \mu\text{m}$. The plot on the left represents the accumulated source function for the excited states transported only by collisions, while the plot on the right demonstrates the comparatively more rapid transport of the radiation, and resulting lower profile. The source term is decreasing at this time, and the radiation transport of the excited states dominates.

3.8. High performance computing

A number of schemes have been implemented to improve the performance of particle simulations [14, 19, 21]. Because particle simulations involve a large number of relatively

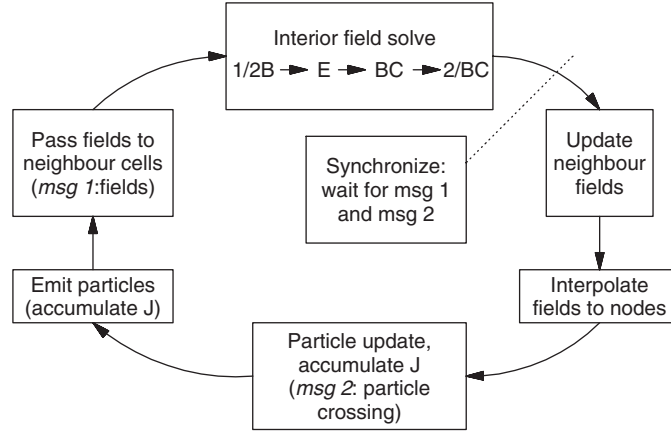


Figure 25. Flow diagram for a parallel electromagnetic PIC code.

brief calculations for each particle, which require primarily localized data, they are amenable to performance enhancement by parallel computing. A scheme for a parallel electromagnetic model is shown in figure 25 [20]. The scheme proceeds in a manner similar to the standard PIC scheme shown in figure 1, except that the field solve is split into interior and neighbouring parts, and messages are sent between neighbours. The first message, labelled msg 1, sends field components required for the solution of Maxwell's equations to neighbouring nodes. The second message, labelled msg 2, contains the particles that have crossed from the mesh contained by one node into that of another. The calculation waits to receive the messages at only the single synchronization point shown, allowing the total load to be balanced without taking into account to the fractional computation time for particles versus fields on each node.

Recent work has examined a number of performance issues on commodity computing platforms separate from parallel computing [4, 5]. Since the ratio of CPU time for writing to reading main memory is about a factor of 4, it is critical to reduce data storage to main memory wherever possible. Since the primary operation in updating particle equations of motion is of the form $y = ax + b$, only two flops are performed until the data are stored in memory, resulting in poor efficiency. Furthermore, a three-dimensional magnetized particle update requires 47 non-contiguous memory accesses (7 particle phase data, plus $4 \times (6 \text{ field elements} + 4 \text{ source terms})$), assuming particle data are stored in separate arrays for each phase variable.

This can be improved by noting that modern commodity processors have been optimized for graphics operations, such as 4-vector multiplication

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & t_x \\ c_{21} & c_{22} & c_{23} & t_y \\ c_{31} & c_{32} & c_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (105)$$

and vector dot products,

$$\mathbf{v} \cdot \mathbf{n} = |\mathbf{v}| |\mathbf{n}| \cos \theta_{vn}. \quad (106)$$

Also, cache thrashing can be reduced by storing particle phase data contiguously, as shown in figure 26, resulting in a performance improvement of up to 100% on Intel x86 processors [5]. The contiguous storage might be as an array of particle structures or a similar mechanism. Finally, it was shown that sorting the particles even every 10–100 time steps, which improves

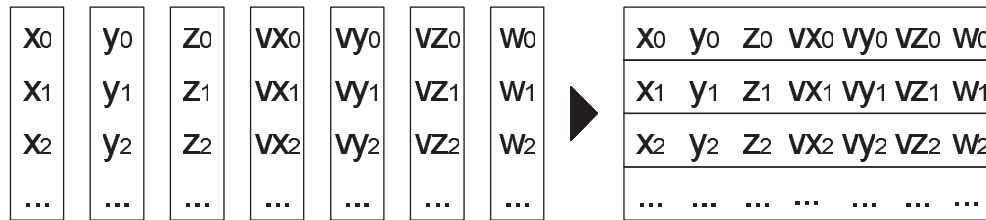


Figure 26. Particle data stored in separate arrays for each element (left) versus contiguous storage for each particle datum (right), which might be an array of structures or classes.

the cache coherency by reusing field data for many consecutive particles within a cell, can result in significant performance improvements.

4. Summary and future challenges

The classical PIC model has well-developed fundamentals. It requires few approximations, and can model the full kinetic, nonlinear and relativistic behaviour of plasmas. Particle models are often used to benchmark more approximate methods, as well as for simulation of a broad array of applications. New models and methods continue to expand the domain of particle methods, as well as improving fidelity and performance. Increasing computer performance also continues to increase the capabilities of particle simulation.

A number of challenges remain for particle methods. Improved models are needed for many device applications, including models for boundary–field and boundary–particle interactions. Integrated models, in which a large range of time and space scales are spanned, as well as regimes with different physics modelled with particles, fluids and hybrids, represent an important area of development in computational plasma physics. Chemistry models continue to be improved, including models for reactive gas mixtures, and multiple excited states and ionization pathways. Performance remains an issue as simulation of larger devices with better fidelity is likely to continue unabated in the foreseeable future. For example, self-consistent simulation of a full fusion reactor will require a multi-petaflop computer, a stage that should be approached within a decade.

Acknowledgments

This work was supported over the years by almost every imaginable agency and organization, including but not limited to AFOSR, AFRL-Edwards, Army Benet Lab., DOE, Hitachi Corp., ONR, NRL, LLNL, Sumitomo Metals, TechX Corp., and many more. Current support from the AFOSR HPM Cathode and Air Breakdown MURI, AFRL-Kirtland, Calabazas Creek Research (AFOSR STTR and DOE SBIR), LLNL Edge SI and the HIF-VNL is gratefully acknowledged. Most of all, the author is grateful to the many members of this community, far too numerous to name, upon whose shoulders he stands, especially the one who mentored so many of us: CK (Ned) Birdsall.

References

- [1] Birdsall C K and Langdon A B 1985 *Plasma Physics via Computer Simulation* (New York: McGraw-Hill)
- [2] Boris J P 1970 Relativistic plasma simulation—optimization of a hybrid code *Proc. 4th Conf. on Numerical Simulation of Plasmas* (Washington, DC) pp 3–67

- [3] Boswell R W and Morey I J 1988 Self-consistent simulation of a parallel-plate rf discharge *Appl. Phys. Lett.* **52** 21–3
- [4] Bowers K J 2003 Accelerating a particle-in-cell simulation using a hybrid counting sort *J. Comput. Phys.* **173** 393–411
- [5] Bowers K J 2003 Speed optimal implementation of a fully relativistic 3d particle push with charge conserving current accumulation on modern processors *18th International Conf. on The Numerical Simulation of Plasmas (N. Falmouth, MA)*
- [6] Buneman O 1959 Dissipation of currents in ionized media *Phys. Rev.* **115** 503–17
- [7] Christlieb A J, Krasny R and Verboncoeur J P 2004 A grid-free treecode field solver for plasma simulations with application to a confined electron column in a penning-malmberg trap *Comput. Phys. Commun.* **164** 306–10
- [8] Dawson J M 1962 One-dimensional plasma model *Phys. Fluids* **5** 445–59
- [9] Dawson J M 1964 Thermal relaxation in a one-species, one-dimensional plasma *Phys. Fluids* **7** 419–25
- [10] Eastwood J W 1991 The virtual particle electromagnetic particle-mesh method *Comput. Phys. Commun.* **64** 252–66
- [11] Gopinath V P, Verboncoeur J P and Birdsall C K 1998 Multipactor electron discharge physics using an improved secondary emission model *Phys. Plasmas* **5** 1535–40
- [12] Green A E S and Sawada T 1972 *J. Atmos. Terr. Phys.* **34** 1719
- [13] Hockney R W and Eastwood J W 1981 *Computer Simulation Using Particles* (New York: McGraw-Hill)
- [14] Blahovec J J Jr, Bowers L A, Luginsland J W, Sasser G E and Watrous J J 2000 3-D PIC simulations of the relativistic klystron oscillator *IEEE Trans. Plasma Sci.* **28** 821–9
- [15] Lawler J E, Parker G J and Hitchon W N G 1993 Radiation trapping simulations using the propagator function method *J. Quant. Spectrosc. Radiat. Transfer* **49** 627–38
- [16] Lawson W S 1989 Particle simulation of bounded 1d plasma systems *J. Comput. Phys.* **80** 253–76
- [17] Lee H J, Kim H C, Yang S S and Lee J K 2002 Two-dimensional self-consistent radiation transport model for plasma display panels *Phys. Plasmas* **9** 2822
- [18] Lee H J and Verboncoeur J P 2001 A radiation transport coupled particle-in-cell simulation: Part I. Description of the model *Phys. Plasmas* **8** 3077
- [19] Liewer P C and Decyk V K 1989 A general concurrent algorithm for plasma particle-in-cell codes *J. Comput. Phys.* **85** 302
- [20] Luginsland J W, Antonsen T A Jr, Verboncoeur J P, Lemke R W, Ludeking L, Mardahl P J, Lin A T, Lau Y Y and Blahovec J D Jr 2001 Computational techniques *High-Power Microwave Sources and Technologies* ed R J Barker and E Schamiloglu (New York: IEEE Press) pp 376–437
- [21] Di Martino B, Briguglio S, Vlad G and Sguazzero P 2001 Parallel PIC plasma simulation through particle decomposition techniques *Parallel Comput.* **27** 295–314
- [22] McDaniel E W 1989 *Atomic Collisions* (New York: Wiley)
- [23] Morse R L and Nielson C W 1971 Numerical simulation of the Weibel instability in one and two dimensions *Phys. Fluids* **14** 830–40
- [24] Nanbu K 1997 Momentum relaxation of a charged particle by small-angle coulomb collisions *Phys. Rev. E* **56** 7314
- [25] Opal C B, Peterson W K and Beaty E C 1971 *J. Chem. Phys.* **55** 4100
- [26] Surendra M, Graves D B and Morey I J 1990 Electron heating in low-pressure rf glow discharges *Appl. Phys. Lett.* **56** 1022–4
- [27] Takizuka T and Abe H 1977 A binary collision model for plasma simulation with a particle code *J. Comput. Phys.* **25** 205–19
- [28] Vahedi V and DiPeso G 1997 Simultaneous potential and circuit solution for two-dimensional bounded plasma simulation codes *J. Comput. Phys.* **131** 149–63
- [29] Vahedi V and Surendra M 1995 Monte Carlo collision model for particle-in-cell method: Application to argon and oxygen discharges *Comput. Phys. Commun.* **87** 179–98
- [30] Verboncoeur J P 2001 A digital filtering scheme for particle codes in curvilinear coordinates *28th IEEE ICOPS (Las Vegas, N.V., 2001)*
- [31] Verboncoeur J P 2001 Symmetric spline weighting for charge and current density in particle simulation *J. Comput. Phys.* **174** 421–7
- [32] Verboncoeur J P, Alves M V, Vahedi V and Birdsall C K 1993 Simultaneous potential and circuit solution for 1d bounded plasma particle simulation codes *J. Comput. Phys.* **104** 321–8
- [33] Verboncoeur J P, Langdon A B and Gladd N T 1995 An object-oriented electromagnetic PIC code *Comput. Phys. Commun.* **87** 199–211
- [34] Villasenor J and Buneman O 1992 Rigorous charge conservation for local electromagnetic field solvers *Comput. Phys. Commun.* **69** 306–16