

DOE/ET-53088-374

IFSR #374

**Particle Simulation in
Curvilinear Coordinate Systems**

M. J. LeBrun and T. Tajima

Institute for Fusion Studies
Department of Physics
The University of Texas at Austin
Austin, Texas 78712

June 1989

Particle Simulation in Curvilinear Coordinate Systems

M.J. LeBrun and T. Tajima
Institute for Fusion Studies
Department of Physics
University of Texas at Austin
Austin, TX 78712

June 7, 1989

Abstract

We present methods for particle simulation of plasmas in a nearly arbitrary coordinate metric and describe a toroidal electrostatic simulation code that evolved from this effort. A Mercier-type coordinate system is used, with a nonuniform radial grid for improved cross-field resolution. A fast iterative method for solving the Poisson equation is employed, and the interpolation/filtering technique shown to be momentum and energy conserving in the continuum limit. Lorentz ion and drift electron species are used. The code has been thoroughly tested for its reproduction of linear and nonlinear physics, and has been applied to the toroidal drift wave problem and its impact on anomalous transport in tokamaks.

1 Introduction

Computer simulation of plasmas plays an important role in understanding the underlying linear and nonlinear physics, as well as improving our ability to predict plasma behavior. Kinetic simulations, for example, have been valuable in the study of two-stream instabilities, collisionless shocks, and double-layer formation, to name a few. In the opposite limit, the physics of an actual magnetic confinement device is exceedingly complex, and may not yield to any single theoretical treatment. Hence we witness a reliance in this case on scaling laws deduced from empirical data, or transport calculations based on approximate growth rates for various instabilities. The modeling of an actual confinement device via simulation therefore represents one of the greatest goals (and challenges) of simulation, for which it is an independent avenue of research and not merely an aid to theory or experiment.

Plasma physics in a general geometry increasingly captures our attention. This occurs primarily due to the importance of the metric on the plasma behavior, which can be particularly relevant for astrophysical plasmas in a general relativistic metric. Additionally, advances in computer technology have made simulations in complex geometries practical. Fluid simulation has been used extensively to model plasmas in geometries approximating that of an actual fusion device [1]. Recent efforts involving magnetohydrodynamic (MHD) codes have established that large-scale tokamak phenomena such as sawteeth can be reproduced by the numerical model [2,3]. These codes are not adequate, however, for investigating the problem of anomalous transport (which is known to involve the microscopic particle dynamics as an essential ingredient). For this we must turn to a kinetic model such as particle simulation, which utilizes a fully self-consistent description of the plasma.

In contrast to the “state of the art” in fluid simulations of the confined plasma, where a relatively realistic geometry is employed, particle simulation is generally performed using a slab coordinate system — corresponding to a small region about one or several rational surfaces. Such a treatment ignores (for example) effects due to trapped particles and mode coupling from the curved magnetic field lines. These and other geometrical effects are difficult to incorporate rigorously into the theory, yet may play a large role in the observed anomalous transport in tokamaks. A generalization of particle simulation methods to more arbitrary metrics is thus warranted. In this work we present methods for particle simulation in a general metric, with

application to the study of tokamak-like plasmas in a toroidal coordinate system and self-consistently determined electrostatic fields.

The choice of a numerical model to represent the physical system must necessarily depend strongly on the known properties of the system. The modes most damaging to the magnetic confinement of a plasma are those with short wavelengths perpendicular to the field, and long wavelengths along the field. We are thereby led to choose an orthogonal coordinate system in which the ambient magnetic field is everywhere orthogonal to the radial coordinate. Spatial derivatives are treated locally in the radial direction, using finite-differencing, while spectral methods are used in the poloidal and toroidal angle variables. This model is generalized further through the adoption of a nonuniform radial mesh, which allows local and global phenomena to be resolved simultaneously.

Discretizing space via coordinates which reflect the structure of the confining magnetic field has many advantages over alternate methods (e.g. see Ref.[4]), such as better cross-field resolution, less numerical diffusion of particles across flux surfaces, and straightforward specification of boundary conditions. We emphasize that our basic model applies to nearly any confined plasma in a curved metric. This includes systems as diverse as a plasma in a mirror confinement device or an ionospheric plasma. It is advantageous, however, to restrict ourselves to systems in which the magnetic fluctuations remain small compared to the background field (which is reasonable for reactor physics of low- β plasmas). If the level of magnetic perturbations is small, the coordinate system remains a good discretization for the length of the run, and adaptive grid methods are not necessary. The particles, while following the magnetic field lines to lowest order, undergo drifts of order ρ/L (Larmor-radius/scale-length) in a physical system; the relative size of magnetic perturbations $\delta B/B$ can reach similar proportions without undue effect on the quality of the simulation. Although such fluctuations can be incorporated into the algorithm (via the Darwin formulation, for example), the remainder of this paper will consider only electrostatic fluctuations.

The use of particles creates some problems that are absent in a fluid algorithm, such as the integration of the particle orbit equations, charge assignment, and charge (force) shaping. Each of these is considered in this work in the context of a general metric. Of particular concern are the effects of the $r = 0$ point (where the Jacobian vanishes) and the nonuniform grid. Common to both fluid and particle codes, however, is the need to efficiently

solve for the field quantities. It is critically important that a fast inversion method exist since the field components are needed at each time step and at many points in space. For elliptic partial differential equations (such as the Poisson equation) in highly symmetric configurations, inversion is usually accomplished via a rapid elliptic solver (RES). The best known of these are based on the fast Fourier transform (FFT) or cyclic reduction. The general metric case, while not directly solvable by an RES, may often be efficiently solved by applying an RES iteratively. We find the toroidal Poisson equation to be amenable to this type of solution, with convergence in less than ten iterations in most cases. In Appendix A we discuss a method of solution for Poisson equation in a more general metric, which is of interest for simulations of moderately high- β tokamak plasmas or other novel configurations of the confining magnetic field.

The rest of the paper is organized as follows. The coordinate system and the discretization of the fields is discussed in Sections 2 and 3, respectively. In Section 4 the field solver and interpolation/filtering scheme is presented, and the energy/momentum conserving properties of the system are analyzed. The particle pushing algorithms are given in Section 5. Tests of the simulation code are given in Section 6, and applications of the code to the toroidal system are shown in Section 7. Summary and additional discussion is given in Section 8.

2 The coordinate system

The geometry we focus on is illustrated in Figure 1. The (r, χ, ζ) coordinate system we use is orthogonal and right-handed, with the surfaces of constant r being nested, concentric circles of revolution (tori). Since we are using the electrostatic approximation, the magnetic fields can be specified in any manner desired, allowing a great deal of flexibility. However, the typical configuration is one in which the magnetic axis lies at $r = 0$, and flux surfaces lie on surfaces of constant r . The term *toroidal* in this work applies to this system only.

We require a simple transition to the limiting cases of cylindrical or slab (Cartesian) geometry. Therefore the *distancelike* coordinates (χ, ζ) are chosen instead of the more usual (θ, ϕ) representation; these lead to straightforward limiting behavior. (Note : we shall often use θ and ϕ interchangeably with χ and ζ in this work—the distinction is important mainly when taking limits.) Specifically, the coordinates χ and ζ are given by

$$\begin{aligned}\chi &= r_0 \theta \\ \zeta &= -R_0 \phi\end{aligned}$$

where R_0 is the major radius, $r_0 = (r_{\min} + r_{\max})/2$, and typically $r_{\min} \sim 0$. The metric components for this system are

$$\begin{aligned}h_r &= 1 \\ h_\chi &= r/r_0 \\ h_\zeta &= R/R_0\end{aligned}\tag{1}$$

where $R = R_0 + r \cos \theta$. Thus the infinitesimal volume element is given by

$$\begin{aligned}J dV &= \frac{r}{r_0} \frac{R}{R_0} dr d\chi d\zeta \\ &= r R dr d\theta d\phi\end{aligned}$$

as expected. For the (χ, ζ) coordinates, the limit to cylindrical or cartesian coordinates is now well-defined :

$$\begin{aligned}\zeta &\rightarrow z && \text{for } R_0 \rightarrow \infty, \text{ and} \\ \chi &\rightarrow y && \text{for } r_0 \rightarrow \infty.\end{aligned}$$

The next step in obtaining proper limiting behavior is to choose a local variable $u = r - r_0$ to represent radial displacement. If the $r_0 \rightarrow \infty$ limit is taken, keeping the radial length $L_r = r_{\max} - r_{\min}$ constant, the local variable u remains finite. Now we introduce the auxiliary (smallness) parameters

$$\begin{aligned}\epsilon_\zeta &\equiv r_0/R_0 \\ \epsilon_\chi &\equiv (L_r/2)/r_0\end{aligned}$$

which measure the strengths of the cylindrical and toroidal effects, respectively (ϵ_ζ is half the inverse aspect ratio). The relationship between the angular coordinates (θ, ϕ) and the present coordinates is expressed in terms of the curvatures

$$\begin{aligned}\kappa_\zeta \equiv 1/R_0 &= \kappa_\chi \epsilon_\zeta \\ \kappa_\chi \equiv 1/r_0 &= \epsilon_\chi / (L_r/2)\end{aligned}$$

where κ_ζ is the curvature along ζ at the magnetic axis, and κ_χ is the curvature along χ at $r = r_0$. The metric components become

$$\begin{aligned}h_u &= 1 \\ h_\chi &= 1 + \kappa_\chi u \\ h_\zeta &= 1 + \epsilon_\zeta (1 + \kappa_\chi u) \cos(\kappa_\chi \chi)\end{aligned}\tag{2}$$

The geometry is then completely specified by the quantities $(L_r, \epsilon_\chi, \epsilon_\zeta)$ in the toroidal system—to take the cylindrical limit one sets $\epsilon_\zeta = 0$ and specifies L_z , and to take the slab limit one sets $\epsilon_\chi = 0$ and specifies L_y . It should be emphasized that by choosing appropriate metric components in Eqs. (2), we can realize nearly any well-behaved geometry using the methods given in this work.

One may represent a field quantity in this coordinate system in terms of the Fourier transform in both χ and ζ since these are (usually) periodic variables; for the electric potential this is given by

$$\Phi(r, k_\chi, k_\zeta) = \sum_{m,n} \Phi(r, \chi, \zeta) \exp(ik_\chi \chi + ik_\zeta \zeta)$$

where the wavevector components are given by

$$\begin{aligned}k_\chi &= \frac{2\pi m}{L_\chi}, & L_\chi &= 2\pi r_0 \\ k_\zeta &= \frac{2\pi n}{L_\zeta}, & L_\zeta &= 2\pi R_0\end{aligned}$$

If the slab limit is taken, this becomes the familiar (x, k_y, k_z) Fourier representation. In the toroidal system this reads

$$\Phi(r, m, n) = \sum_{m, n} \Phi(r, \theta, \phi) \exp(im\theta - in\phi).$$

We have also considered a configuration which employs a radial “slice” (or annulus), that is, the inner boundary $r = r_{\min}$ is far from zero. The use of such a model is advantageous due to the fine resolution in r that one may obtain. However, having a radial boundary where the density is maximum limits the applicability of such a model. Our tests have shown that significant modification of the distribution occurs near the boundary in a toroidal system; this is a consequence of first-order particle drifts into the boundaries (banana orbits), primarily of the ion species. The outer boundary can in principle be ignored because the density can be made arbitrarily small there, but the inward boundary remains a concern. We believe that the “slice” model may yet prove useful with some modifications to the algorithm and boundary handling, but this remains untested. We therefore limit our discussion to the model in which essentially the entire radial extent is modeled (achieving locality through the nonuniform radial grid).

3 Representation of the field quantities

3.1 The nonuniform radial grid

3.1.1 Motivation

In any numerical calculation, the primary motivation for using a nonuniform mesh is to more closely match the sampling method (i.e. grid) with the behavior of the solution. This idea was perhaps first used in fluid simulations. For example, in laminar flow around solid bodies, the character of the flow in the boundary layer varies much more rapidly than in the interior of the flow region. Thus, there is the desire to increase resolution in the boundary layer by concentrating grid points there, and spread out grid points elsewhere, to minimize computational expense. The term *boundary-fitted coordinates* is used in fluid dynamics to describe the generated-grid system for studying flow around a solid object [5]. In plasma simulation, this boundary is manifested by the magnetic field. Generated grids which conform to the shape of the magnetic field are commonly seen in modern MHD simulations.

The use of nonuniform or generated grids in particle simulation has been rare, however. This may be due to the fundamental differences between particle and fluid simulation. First, particle simulations have been primarily aimed at *local* phenomena, and therefore the exact specification of the magnetic fields is less important than in the global, fluid simulations. It is this local emphasis that permits the adoption of slab geometry for many particle simulations. Second, existing particle simulations often rely on k-space representations [6] for the solution of the fields, making it difficult to incorporate even a single, stretched variable. And third, the resolution in a particle simulation *is not determined by the grid alone*. In a particle simulation we must keep the number of particles per cell appreciable in order to accurately model plasma behavior there, which limits the amount of grid stretching that may be employed.

This is not to say that nonuniform grids are ill-suited to particle simulations. Even local problems often show multiple scales of length, and a nonuniform grid could help considerably. For example, the present code has been employed for study of drift waves in a tokamak-like plasma; the nonuniform grid in r gives good resolution near the mode rational surfaces, while allowing the ion resonance surfaces to remain within the simulation region.

Another example application is in the study of tearing modes—the resolution immediately surrounding the tearing layer could be much improved by using a nonuniform mesh.

The implementation of a nonuniform grid in one variable (or grid stretching) usually requires the adoption of a finite-difference formulation. (See Ref. [7] for a discussion of nonuniform grids in a purely k -space representation.) In our case, radial finite-differencing is necessary for solution of the toroidal field equations, and the addition of the nonuniform radial grid is straightforward. Further, the finite-difference approach seems more logical when the plasma variation in the coordinate is highly nonuniform (see Ref. [8] for more discussion of this point).

Finally, the need to maintain acceptable numbers of particles per cell represents a definite limitation. However, the effect of this restriction is minimal when the region of the greatest number of particles coincides with the region of maximum resolution. For a simulation in slab geometry that includes a density profile, this location is at the inward boundary. However, we often want the region of interest to lie in the center of our simulation “window” to minimize boundary effects, in which case the proportion of particles in the central cells becomes extremely small. The situation improves markedly when one moves to cylindrical geometry (therefore toroidal geometry as well), as a result of the uniform-grid cell area being proportional to r . In cylindrical geometry, the density profile should be flat at $r = 0$. If we also assume that the density vanishes faster than $1/r$ for large r , and a uniform grid, then the number of particles per cell will be nearly zero at the origin, rise to a peak, and then decay. Careful choice of the nonuniform grid allows a nearly uniform distribution of particles, which is the ideal situation. In this case, the maximum resolution will be governed by the sharpness of the plasma profile. An example density profile (gaussian), grid spacing, and number of particles per cell in a cylindrical coordinate system is given in Figure 2. In Figure 3 we show an example (r, θ) grid configuration employing 64 radial gridpoints (nonuniformly distributed), and 32 theta gridpoints.

3.1.2 Local theory

In order to rigorously justify the use of a nonuniform grid, we consider the local properties of such a system. To start, we examine a uniform grid system of grid spacing Δ . The centered, second order finite-difference representation

of the first and second derivatives on a uniform grid are given by

$$\frac{f_{i+1} - f_{i-1}}{2\Delta} = f'_i + \frac{\Delta^2}{6} f'''_i + \dots \quad (3)$$

$$\frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta^2} = f''_i + \frac{\Delta^2}{12} f^{IV}_i + \dots \quad (4)$$

where f is an arbitrary function. The rationale for using these approximations is often based on their second-order behavior, but this is only an asymptotic relation between local error and grid spacing, and leaves out the effects of the higher-order derivatives of f . It is useful to assume a particular form for f and re-examine the accuracy of these equations. For the phenomena of interest here, solutions have an oscillatory character, so it is reasonable to express f in terms of its Fourier harmonics. Then f and f' are of the form

$$\begin{aligned} f(x) &= \sum f_k e^{ikx} \\ f'(x) &= \sum f_k ik e^{ikx} \end{aligned}$$

and finite difference approximation to f' is given by

$$\begin{aligned} f'(x) &\approx (f_{i+1} - f_{i-1}) / 2\Delta \\ &= \sum f_k e^{ikx} i (\sin k\Delta) / \Delta \end{aligned}$$

which equals the exact result in the limit of small $k\Delta$. The result for the second derivative is similarly given by

$$\begin{aligned} f''(x) &\approx (f_{i+1} - 2f_i + f_{i-1}) / \Delta^2 \\ &= -\sum f_k e^{ikx} (\sin(k\Delta/2))^2 / (\Delta/2)^2 \end{aligned}$$

Clearly, the finite-difference approximation does well for long wavelength modes, but becomes inaccurate in the neighborhood of $k\Delta = \pi/2$. For $k\Delta = \pi/4$, the first derivative approximation is accurate to within about 10 percent, and the second derivative approximation to within 5 percent. Particle codes usually employ k-space filters to eliminate the high mode numbers (discussed in section 4.2); the weakening effect due to such a filter will often far exceed that of the finite-difference operator. For example, a gaussian filter, for a particle the size of the grid spacing, halves the signal strength at $k\Delta = \pi/4$.

Thus the second-order finite difference expressions will usually be sufficiently accurate for use in a particle code.

We now consider a one-dimensional, nonuniform (stretched) grid. Although the finite-difference approximation to the first derivative generalizes easily :

$$\frac{df}{dx} \simeq \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}}, \quad (5)$$

expressions for the second derivative approximation and truncation error involve considerable complexity. It is helpful to adopt the notation

$$\begin{aligned} \bar{\Delta}_i &\equiv (\Delta_{i+1/2} + \Delta_{i-1/2})/2 \\ \delta_i &\equiv (\Delta_{i+1/2} - \Delta_{i-1/2}) \\ \gamma_i &\equiv \delta_i/(2\bar{\Delta}_i) \end{aligned} \quad (6)$$

where $\bar{\Delta}_i$ is the mean cell size and δ_i is the change in grid spacing at the i th grid point. The quantity γ_i is approximately one half the rate of change of the grid spacing, a useful measure of the nonuniformity of the grid. With these definitions, the finite difference approximation for the first derivative is

$$\frac{f_{i+1} - f_{i-1}}{2\bar{\Delta}_i} = f'_i + \gamma_i \bar{\Delta}_i f''_i + (1 + 3\gamma_i^2) \frac{\bar{\Delta}_i^2}{6} f'''_i + \dots \quad (7)$$

and for the second derivative is

$$\frac{(1 - \gamma_i) f_{i+1} - 2f_i + (1 + \gamma_i) f_{i-1}}{\bar{\Delta}_i^2 (1 - \gamma_i^2)} = f''_i + \gamma_i \frac{2\bar{\Delta}_i}{3} f'''_i + (1 + 3\gamma_i^2) \frac{\bar{\Delta}_i^2}{12} f^{IV}_i + \dots \quad (8)$$

which essentially match expressions given elsewhere [9]. These expressions show clearly the effect of the stretched variable, and are different from their uniform grid counterparts in an important qualitative way—these approximations are no longer formally second-order accurate in the grid spacing. The reason for this loss of formal accuracy is that for a nonuniform grid, these approximations to the first and second derivatives are *no longer perfectly centered*, resulting in the first-order Δ_i term in the truncation error. For extremely rapidly varying grid size, γ_i can become of order unity, which causes the approximation to be first order accurate only (in this case, the finite-difference approximations have become essentially one-sided). It is clear

that by changing the grid spacing sufficiently slowly, γ_i can be neglected and formal second order accuracy will be retained.

Eqs. (7)–(8) have the disadvantage of complexity and difficult extension to higher order. It is usually much simpler to express derivatives in terms of a transformed variable. We define the transformed coordinate $\xi = \xi(x)$ where the $\xi(x)$ is our stretching-function, and grid-points occur in ξ -space at intervals of constant $\Delta\xi$ (arbitrarily set to 1). Expressing the derivatives of f in terms of ξ , we have

$$\frac{\partial f}{\partial x} = \frac{\partial \xi}{\partial x} \frac{\partial f}{\partial \xi} \quad (9)$$

$$\frac{\partial^2 f}{\partial x^2} = \left(\frac{\partial \xi}{\partial x} \right)^2 \frac{\partial^2 f}{\partial \xi^2} + \frac{\partial^2 \xi}{\partial x^2} \frac{\partial f}{\partial \xi}. \quad (10)$$

Since the partial derivatives of $\xi(x)$ are assumed to be known, all that remains is to replace $\partial f / \partial \xi$ and $\partial^2 f / \partial \xi^2$ by their corresponding *uniform grid* finite difference representations. This method is equivalent to the direct use of Eqs. (7)–(8) despite obviously being second order accurate in $\Delta\xi$ (for a good discussion of these issues, see Refs. [10,11]).

A more meaningful measure of accuracy is obtained through Fourier-analysis of the function f , whereupon the finite-difference approximation to the first derivative becomes

$$\begin{aligned} f'(x) &\approx (f_{i+1} - f_{i-1}) / (2\bar{\Delta}_i) \\ &= \sum f_k e^{ikx} i e^{ik\delta_i/2} (\sin k\bar{\Delta}_i) / \bar{\Delta}_i. \end{aligned}$$

This is virtually equivalent to the uniform grid result, the main difference being the presence of a phase factor $e^{ik\delta_i/2}$. We obtain an accuracy condition on the nonuniformity of the grid, relevant to particle simulation, by requiring the argument of this phase factor to be small at the “cut-off” value $k\bar{\Delta}_i \sim \pi/2$. Given $k\delta_i/2 \ll \pi/2$, we then have

$$\gamma_i \ll 1 \quad (11)$$

for our nonuniform grid condition.

In the present work, the transformation function approach has been used exclusively. This approach has the advantages of smoothness and simplicity. For the nonuniform grids employed, γ_i is typically smaller than 0.1 (but is a

function of position), giving an accurate finite-difference representation for most of the wavenumber spectrum. The error estimation argument followed here is still largely *ad hoc*, and may not give the best results for highly nonuniform situations. In such situations the choice of the optimum grid is mostly an educated guess and highly problem-dependent; it is probably best to be conservative in the choice of grid parameters, and keep γ_i small.

3.1.3 Construction of stretching function

We now wish to construct a stretching function according to some prescribed grid spacing. Denoting the grid spacing by $\Delta(x)$, we have the relation

$$\begin{aligned}\Delta(x) &\equiv \frac{\Delta x}{\Delta \xi}(x) \\ &\simeq \frac{1}{\partial \xi / \partial x}\end{aligned}\tag{12}$$

which then *defines* the stretching function

$$\xi(x) \equiv \int \frac{dx}{\Delta(x)}.\tag{13}$$

The resulting grid spacing does not precisely match the function $\Delta(x)$ because of discreteness effects, but is usually close. The first restriction on $\Delta(x)$ is that it yield a $\xi(x)$ with continuous first and second partial derivatives (this will be true if $\Delta(x)$ and $\Delta'(x)$ are continuous and nonzero). These derivatives are required to solve the Poisson equation. In general, this condition is not difficult to satisfy.

The second restriction we make on the generation of the stretching function is that $\xi(x)$ be *simple*, that is, $\xi(x)$ should be analytically expressible with a minimum of special function evaluations. This simplifies the transformation of the particle positions x_i to the transformed coordinate in both the charge accumulation and the field interpolation phases. The result is a reasonably fast, vectorized loop. For increased generality, one could solve for $\xi(x)$ at discrete values only, then interpolate to get ξ as a function of the particle coordinates. This type of loop vectorizes on computers that support gather/scatter-type vectorization (Cray-2, Cray X-MP 48), and may be faster than the loop employing the exact stretching function.

$\Delta(x)$	$\xi(x)$
$c_1 \cosh^2(c_2 \tilde{x})$	$c'_1 + c'_2 \tanh(c'_3 \tilde{x})$
$c_1 (c_2 + c_3 \tilde{x}^2) / (c_2 + \tilde{x}^2)$	$c'_1 + c'_2 \tilde{x} + c'_3 \tan^{-1}(c'_4 \tilde{x})$

Table 1: Examples of stretching functions ($\tilde{x} \equiv x - x_a$)

For the type of grid we are interested in, $\Delta(x)$ has a minimum at some point $x = x_a$, and increases outward from that point (possibly reaching some limit). We give two such functions in Table 1 that satisfy the above restrictions. Each function has only one special function evaluation in the expression for $\xi(x)$. It is this evaluation that is the most time-consuming part of the entire method, and timing tests show an increase in total cpu-time of only a few percent for a typical configuration. The second function in Table 1 is the one employed in this work—it allows choice of a limiting (maximum) value for the grid spacing, as well as the degree of grid stretching. The main disadvantage with this stretching function is that an exact expression for $x(\xi)$ is not known. However, $x(\xi)$ is only needed at grid points for diagnostics; these values are obtained via a simple root-finder routine.

3.1.4 Discussion

Here we focus on two remaining concerns, of which the first considers the meaning of particle *shapes* in a simulation with a nonuniform grid. The concept of a particle shape in a particle simulation enters primarily through the grid. When one interpolates to the nearest two to three gridpoints in each direction, the response of the system indicates an effective particle size on the order of a cell-width. This response is often modified by a filter (or finite-size particle shape factor). The use of a nonuniform grid does not alter this appreciably. If we modify our filter so that it filters only locally, then the relationship between the effective particle size and the cell size is maintained. This is accomplished using a technique we refer to as *digital filtering*, discussed in section 4.2. As the cell-width changes due to the nonuniform grid, so does the effective particle width. This is a natural effect, and is an essential property to have true control over the resolution.

If particles can easily move along a given coordinate, a nonuniform grid in

that direction is not appropriate. The discretization method must be chosen in accordance with the boundary conditions and the problem at hand. In the present work, the use of a nonuniform grid is appropriate in the direction of the density gradient, and perpendicular to the dominant magnetic field direction.

The second concern we shall touch on is the need to retain an adequate number of particles per cell (in the region of interest) as the cell size is decreased. As discussed previously, this problem occurs because we are not free to arbitrarily specify the location of particles when a given density variation is imposed. In a cylinder or torus we see a reduction of this effect, and can even lead to nearly equal numbers of particles per cell (cf. Fig. 2). Therefore we see a significant gain in resolution in these systems through the adoption of a moderately nonuniform radial grid. We next seek modifications of the algorithm which will allow grids of stronger nonuniformity and more practical application to slab geometry.

Since the major restriction on the particle number concerns the representation of the background density gradient, one option is to remove the background distribution via a multiple distribution approach. That is, we split the distribution f arbitrarily into two parts—one that is stationary (f_0), and the remainder (δf) which evolves and is discretized in phase space via the weighted “particle” species. We refer to this as the δf or weighted particle method [12,13].

Each “particle” in the simulation thus comes to represent a bit (δf_j) of the perturbed distribution at that particular point in phase space. This discretization can be expressed formally by

$$\delta f = \sum_j \delta f_j(\mathbf{x}_j(t), \mathbf{v}_j(t), t) \delta(\mathbf{x} - \mathbf{x}_j(t)) \delta(\mathbf{v} - \mathbf{v}_j(t)). \quad (14)$$

Operating on this representation of the perturbed distribution function by d/dt , we obtain (after cancellation of terms)

$$\frac{d\delta f}{dt} = \sum_j \frac{d\delta f_j}{dt} \delta(\mathbf{x} - \mathbf{x}_j(t)) \delta(\mathbf{v} - \mathbf{v}_j(t)), \quad (15)$$

with the particle trajectories given by (for a Vlasov plasma)

$$\frac{d\mathbf{x}_j}{dt} = \mathbf{v}_j \quad (16)$$

$$\frac{d\mathbf{v}_j}{dt} = \frac{q}{m} \left(\delta \mathbf{E} + \frac{\mathbf{v}}{c} \times \mathbf{B} \right), \quad (17)$$

and the usual relationship between $\delta \mathbf{E}$ and δf . The quantity δf_j thus evolves according to

$$\frac{d\delta f_j}{dt} = - \left. \frac{df_0}{dt} \right|_{(\mathbf{x}_j, \mathbf{v}_j)}, \quad (18)$$

where the total distribution satisfies $df/dt = 0$. For full dynamics, the evolution equation for δf_j becomes

$$\frac{d\delta f_j}{dt} = - \left[\mathbf{v} \cdot \nabla f_0 + \frac{q}{m} \left(\mathbf{E} + \frac{1}{c} \mathbf{v} \times \mathbf{B} \right) \cdot \nabla_{\mathbf{v}} f_0 \right] \Big|_{(\mathbf{x}_j, \mathbf{v}_j)}, \quad (19)$$

and for drift dynamics,

$$\frac{d\delta f_j}{dt} = - \left[(v_{\parallel} \mathbf{b} + \mathbf{v}_d) \cdot \nabla f_0 + \frac{q}{m} \mathbf{E} \cdot \nabla_{\mathbf{v}} f_0 \right] \Big|_{(\mathbf{x}_j, \mathbf{v}_j)}, \quad (20)$$

where \mathbf{v}_d is the perpendicular drift.

For modelling quasi-stationary phenomena one chooses f_0 to be the equilibrium distribution, with the ratio $\delta f/f$ given at startup (and usually small). The separation of the distribution into background and perturbed part in this case reduces noise levels dramatically [14], thus improving the ability to observe subtle effects. We emphasize that no modification of the nonlinear physics has been made. The main limitations of the method are (a) particles must be loaded uniformly in phase space for accurate charge density calculation, and (b) much of the advantage over conventional simulation is lost if $\delta f/f \sim 1$ is reached. A more complete treatment can be found in Ref. [14].

Finally, we briefly mention another method which may help improve resolution in particle simulation, which we refer to as nonuniform *number* (or mass) weighting. The crux of this method is to allow each simulation particle to represent a varying number of actual particles, adjusted in order to increase the effective resolution. This method can be regarded as a special case of the above weighted particle method in which we now fix the weight in time by representing the entire distribution via the particle species. For example, we might adopt a small grid spacing in a critical region through the use of the nonuniform grid, while retaining an acceptable number of particles per cell by reducing the weight given to particles accordingly. The number weighting

method represents a logical extension to the macroparticle concept, in which one envisions a simulation particle as representing an aggregate of smaller particles, all moving together.

Two effects limit the usefulness of this scheme : (a) interchange of differently weighted particles, and (b) enhanced fluctuations in regions of relatively large particle weight. Both of these effects limit how strong the variation of particle weight may be. Naturally, diffusion of differently weighted particles renders this scheme ineffective, so we restrict the variation of particle weight to perpendicular to the magnetic field (radial). Test runs have shown reasonable behavior for cross-field variation of the particle weight on the order of 2-4.

Combining these techniques in a simulation code may be particularly effective at reducing noise while improving resolution. At this time both methods remain experimental, however, and are not used in the remainder of this work.

3.2 The mode expansion

While grid methods have been applied with much success to the two-dimensional simulation of plasmas, limitations in computer memory have prevented a wide-scale extension to three dimensions. This led to the development of the mode expansion method [4], which Fourier-decomposes the field quantities in the dominant direction of the magnetic field, usually taken as the (z) coordinate. Typically, the representation of the fields becomes

$$\begin{aligned}\rho(x, y, z) &= \sum \rho_n(x, y) e^{ik_z z} \\ \Phi(x, y, z) &= \sum \Phi_n(x, y) e^{ik_z z} \\ \mathbf{E}(x, y, z) &= \sum \mathbf{E}_n(x, y) e^{ik_z z},\end{aligned}\tag{21}$$

where $k_z = 2\pi n/L_z$ and the sum goes from $n = -N$ to $n = N$, N being the number of modes in the z -direction. If we assume symmetry in the third dimension, the field components for each mode can be calculated independently, and later summed in the interpolation phase. Since the toroidal system has azimuthal symmetry, the above equations are applicable to the toroidal algorithm simply by associating (x, y, z) with (r, χ, ζ) .

The main drawback to the mode-expansion method is that the mode-particle loop for accumulation and interpolation involves considerable compu-

tational expense, easily dominating the rest of the code. In the interpolation phase, the mode expansion requires roughly n_z trigonometric evaluations per particle, where n_z is the number of modes in the calculation. On the other hand, the grid method requires only a small number of relatively inexpensive algebraic evaluations for each particle. Therefore, on large memory machines (such as the Cray-2) a full three-dimensional grid representation is usually the more computationally efficient method.

The high accuracy of the mode expansion nevertheless makes it preferable to a grid for some low dimensionality cases. One example of this is the “high- n ” configuration used in Section 7.2 for study of the toroidal drift wave problem. In this case, a single ($n = 9$) or multiple modes were employed, at considerably less computational cost (and higher accuracy) than a grid formulation. Furthermore, for very long system lengths along the magnetic field, grid representations are susceptible to what is known as the “aliasing-instability” [15]. This unphysical effect stems from sampling error when the thermal velocities (λ_D/Δ) are small, and can be avoided in practice by adopting a higher-order spline interpolation scheme [16]. When using a mode expansion in the third dimension, the aliasing instability can be avoided.

Both the grid and mode representations have proven useful. These are currently both implemented in the simulation code, and selected via a switch at the preprocessor level.

4 Field Solver

4.1 Poisson equation

For the electrostatic model, the only field equation that must be solved is the Poisson equation

$$\nabla^2 \Phi = -4\pi\rho \quad (22)$$

$$\mathbf{E} = -\nabla\Phi. \quad (23)$$

Given the metric coefficients for our system (Eqs. (1)), the Laplacian is given by

$$\nabla^2 = \nabla_c^2 + \mathcal{C} \quad (24)$$

with ∇_c^2 and \mathcal{C} given by

$$\nabla_c^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{r_0^2}{r^2} \frac{\partial^2}{\partial \chi^2} + \frac{\partial^2}{\partial \zeta^2} \quad (25)$$

$$\mathcal{C} = \frac{\cos \theta}{R} \frac{\partial}{\partial r} - \frac{\sin \theta}{R} \frac{r_0}{r} \frac{\partial}{\partial \chi} + \frac{R_0^2 - R^2}{R^2} \frac{\partial^2}{\partial \zeta^2}. \quad (26)$$

Here ∇_c^2 is the cylindrical Laplacian, and \mathcal{C} represents the toroidal corrections. The cylindrical Laplacian can be inverted by a standard technique—finite differencing in r and applying the recurrence solution for a tridiagonal matrix. However, the toroidal Laplacian as a whole is *not* directly invertible. This leads us to introduce an approximate method, based on the observation that

$$\nabla^2 = \nabla_c^2(1 + \mathcal{O}(\epsilon_\zeta)), \quad (27)$$

that is, the correction terms are formally of order ϵ_ζ compared to the cylindrical Laplacian. Since ϵ_ζ tends to be small ($\lesssim 0.2$), we expand the potential in powers of ϵ_ζ ,

$$\Phi = \sum_n \epsilon_\zeta^n \psi^n, \quad (28)$$

and substitute into the Poisson equation, obtaining

$$\sum_n \epsilon_\zeta^n \psi^n = \sum_n (\nabla_c^2)^{-1} [-4\pi\rho - \mathcal{C}(\epsilon_\zeta^n \psi^n)]. \quad (29)$$

Equating terms of the same order in ϵ_ζ yields the following chain of equations (recall $\mathcal{C} \sim \mathcal{O}(\epsilon_\zeta)$) :

$$\begin{aligned}\psi^0 &= (\nabla_c^2)^{-1} [-4\pi\rho] \\ \psi^1 &= (\nabla_c^2)^{-1} \mathcal{C} [-\psi^0/\epsilon_\zeta] \\ &\vdots \\ \psi^n &= (\nabla_c^2)^{-1} \mathcal{C} [-\psi^{n-1}/\epsilon_\zeta].\end{aligned}$$

The sum is truncated upon reaching the desired order in ϵ_ζ for convergence. This method is closely related to the fixed point iteration scheme

$$\Phi^{p+1} = (1 - \alpha)\Phi^p + \alpha(\nabla_c^2)^{-1} [-4\pi\rho - \mathcal{C}(\Phi^p)], \quad (30)$$

where α is a relaxation parameter, typically on the order of one, and p represents the iteration count. When no relaxation is used ($\alpha = 1$), the two methods are identical. In practice, the code uses Eq. (30) in the field solver. The convergence criterion for the potential is

$$\frac{\|\Phi^{p+1} - \Phi^p\|}{\|\Phi\|} \leq \epsilon,$$

with $\epsilon \sim 10^{-5}$. Since $\|\Phi\|$ is unknown, the larger of $\|\Phi^{p+1}\|$ or the norm of a previous n-mode is used. (The outer loop over toroidal mode number starts with lowest mode numbers first, which are also the strongest modes due to the filtering effect at high n .) This prescription avoids excessive computation for high mode-number, weak modes.

The inversion of the cylindrical Laplacian is performed in (ξ, k_x, k_ζ) space, where ξ is the transformed coordinate for the nonuniform grid. The finite difference representations of the radial derivatives at the i th gridpoint become

$$\begin{aligned}\left(\frac{\partial\Phi}{\partial r}\right)_i &= \left(\frac{\partial\xi}{\partial r}\frac{\partial\Phi}{\partial\xi}\right)_i \\ &\approx \left(\frac{\partial\xi}{\partial r}\right)_i \left(\frac{\Phi_{i+1} - \Phi_{i-1}}{2}\right)\end{aligned} \quad (31)$$

for the first derivative and

$$\left(\frac{\partial^2\Phi}{\partial r^2}\right)_i = \left[\left(\frac{\partial\xi}{\partial r}\right)^2 \frac{\partial^2\Phi}{\partial\xi^2} + \frac{\partial^2\xi}{\partial r^2} \frac{\partial\Phi}{\partial\xi}\right]_i$$

$$\approx \left(\frac{\partial \xi}{\partial r} \right)_i^2 (\Phi_{i+1} - 2\Phi_i + \Phi_{i-1}) + \left(\frac{\partial^2 \xi}{\partial r^2} \right)_i \left(\frac{\Phi_{i+1} - \Phi_{i-1}}{2} \right) \quad (32)$$

for the second derivative. These employ the usual two and three-point centered difference formulas on the transformed grid. The cylindrical Poisson equation at the i th gridpoint then has the form

$$\alpha_i \Phi_{i+1} + \beta_i \Phi_i + \gamma_i \Phi_{i-1} = s_i \quad (33)$$

with the coefficients given by

$$\begin{aligned} \alpha_i &= \left[\left(\frac{\partial \xi}{\partial r} \right)^2 + \frac{1}{2} \frac{\partial^2 \xi}{\partial r^2} + \frac{1}{2r} \frac{\partial \xi}{\partial r} \right]_i \\ \beta_i &= - \left[2 \left(\frac{\partial \xi}{\partial r} \right)^2 + \frac{r_0^2}{r^2} k_x^2 + k_\zeta^2 \right]_i \\ \gamma_i &= \left[\left(\frac{\partial \xi}{\partial r} \right)^2 - \frac{1}{2} \frac{\partial^2 \xi}{\partial r^2} - \frac{1}{2r} \frac{\partial \xi}{\partial r} \right]_i \end{aligned} \quad (34)$$

and s_i represents the source term. The direct recurrence solution for this type of equation is well known in finite-difference fluid simulation (see Refs. [17, 18]). We look for a recurrence relation of the form

$$\Phi_{i-1} = x_i \Phi_i + y_i, \quad (35)$$

where x and y are to be determined (the roles of i and $i-1$ are often reversed, yielding similar results). Substitution into the field equation (Eq. (33)) gives

$$\alpha_i \Phi_{i+1} + \beta_i \Phi_i + \gamma_i (x_i \Phi_i + y_i) = s_i,$$

which can be rearranged to yield

$$\Phi_i = -\frac{\alpha_i}{\beta_i + \gamma_i x_i} \Phi_{i+1} + \frac{s_i - y_i \gamma_i}{\beta_i + \gamma_i x_i}.$$

Since this is of the same form as Eq. (35), the coefficient terms are given by

$$x_{i+1} = \frac{-\alpha_i}{\beta_i + \gamma_i x_i} \quad (36)$$

$$y_{i+1} = \frac{s_i - y_i \gamma_i}{\beta_i + \gamma_i x_i}, \quad (37)$$

which are calculated in a forward scan. The field component (Φ_i) is subsequently obtained by a backward scan, given by Eq. (35). The only work remaining is to determine the starting values for each scan; these are determined by the boundary conditions on the potential. This is complicated by the location of the boundaries—the only case usually considered (as in the above references) is that of Dirichlet boundary conditions with the boundaries on the end grid points. For the current system, with the boundaries between grid points, we represent the parity at the boundaries by

$$\begin{aligned}\Phi_{N+1} &= \eta \Phi_N \\ \Phi_0 &= \eta \Phi_1,\end{aligned}$$

where the cells in the system number from 1 to N . The virtual grid-points at $i = 0$ and $i = N + 1$ only serve to define the parity at the boundary, which is even parity for $\eta = +1$ ($\Phi' = 0$), and odd parity for $\eta = -1$ ($\Phi = 0$). The field equation at the $i = 1$ grid point is then

$$s_1 = (\eta\alpha_1 + \beta_1)\Phi_1 + \gamma_1\Phi_2,$$

which can be rearranged to the form of the recurrence relation (Eq. (35)) :

$$\Phi_1 = -\frac{\gamma_1}{\eta\alpha_1 + \beta_1}\Phi_2 + \frac{s_1}{\eta\alpha_1 + \beta_1}.$$

The recurrence relation then requires that

$$\begin{aligned}x_2 &= \frac{-\gamma_1}{\eta\alpha_1 + \beta_1} \\ y_2 &= \frac{s_1}{\eta\alpha_1 + \beta_1}\end{aligned}$$

for the starting values of the coefficient arrays in the forward scan. For the backward scan to solve for the potential, a similar calculation gives

$$\Phi_N = \frac{s_N - \gamma_N y_N}{\beta_N + \eta\alpha_N + \gamma_N x_N}$$

as the starting value.

The last question with the tridiagonal inversion concerns execution speed. Since this is a direct solution, it is much faster than general matrix inversion

techniques. However, this solution is a marching-process in i , inhibiting straightforward vectorization. We therefore vectorize in the perpendicular direction (χ) by looping over the second coordinate in the inner-most loop, which requires two-dimensional arrays for the coefficients of the recurrence relation (35). The result is a fast and accurate solution; using it as the core part of an iteration method is reasonable.

For $\epsilon_\zeta \lesssim 0.2$ (corresponding to an aspect ratio of 2.5), the method usually converges to at least five decimal places accuracy in 4-6 iterations for a typical plasma profile. The Poisson solver is structured so that each toroidal mode is calculated separately, which is possible due to the azimuthal symmetry. This allows numerous temporary arrays to be generated to assist in the inversion. The independent calculation of each toroidal mode also forces the adoption of a “separated” (k_χ, k_ζ) mode representation—the Fourier transforms in χ , required for the toroidal corrections, must be independent of the transforms in ζ . In the usual mode representation (i.e. FFT2 structure), the modes cannot be transformed independently.

The electric field components are calculated from the potential, using the toroidal metric :

$$\begin{aligned} E_r &= -\frac{\partial \Phi}{\partial r} \\ E_\chi &= -\frac{r_0}{r} \frac{\partial \Phi}{\partial \chi} \\ E_\zeta &= -\frac{R_0}{R} \frac{\partial \Phi}{\partial \zeta} \end{aligned} \tag{38}$$

For the radial field, we again apply the two-point finite difference approximation for the derivative to get

$$(E_r)_i = \left(\frac{\partial \xi}{\partial r} \right)_i \left(\frac{\Phi_{i+1} - \Phi_{i-1}}{2} \right) \tag{39}$$

The χ and ζ derivatives are simply multiplications in (r, k_χ, k_ζ) space. The r_0/r multiplication for the E_χ component is straightforward, but the R_0/R multiplication for the E_ζ component requires transformation to (r, θ) space because of the θ dependence in R (and must be transformed back again, for filtering).

4.2 Interpolation and Filtering

In this section we discuss interpolation and filtering methods for an arbitrary coordinate system that are accurate, easily implemented, and independent of the metric wherever possible. The three stages of the charge accumulation are considered first—interpolation of the density to the grid, transformation weighting, and filtering.

The initial interpolation is done on a computational grid, that is, with no reference to the actual metric (aside from constant scale factors). The resultant pseudo-charge density (σ) is identical to that obtained in a slab coordinate system. This procedure views the coordinate transformation as a property of the *grid*, freeing us from the difficulties in considering particle “shapes” in arbitrary coordinate systems. Additionally, this procedure enables us to employ the same interpolation routines independent of geometry, which simplifies the implementation considerably.

The actual interpolation methods here fall into the category of area-weighting methods, which we briefly summarize. Letting x_j denote the position of the j th particle, x_g an arbitrary grid-point, and $\delta = x_j - x_g$, the particle weight for that grid point will be given by

$$w(x_g) = \begin{cases} 1 - \frac{\delta}{\Delta} & |\delta| \leq \Delta \\ 0 & \text{otherwise,} \end{cases} \quad (40)$$

for linear weighting (*aka* cloud-in-cell), and

$$w(x_g) = \begin{cases} \frac{3}{4} - \left(\frac{\delta}{\Delta}\right)^2 & |\delta| \leq \frac{\Delta}{2} \\ \frac{1}{2} \left(\frac{3}{2} - \frac{\delta}{\Delta}\right)^2 & \frac{\Delta}{2} \leq |\delta| \leq \frac{3\Delta}{2} \\ 0 & \text{otherwise,} \end{cases} \quad (41)$$

for quadratic weighting (*aka* triangular-shaped-cloud) [18]. In the toroidal nonuniform grid system the radial particle coordinates u_i are transformed to the stretched coordinates ξ_i for the interpolation.

Linear weighting is second-order accurate in the grid spacing and the interpolated function is continuous as the particle crosses a cell boundary. Quadratic weighting is also second-order accurate, but both the interpolated function and its first derivative are continuous as the particle crosses a cell

boundary. The additional smoothness of quadratic weighting results in less error due to undersampling—the aliased wavenumber contributions are much smaller than in linear weighting [8, pages 168-170]. The simulation is thus better able to tolerate small values of λ_D/Δ without numerical heating, and needs less k-space filtering. This is important mainly along the direction of the magnetic field (long dimension). Therefore, we typically use quadratic weighting in the ζ coordinate for a 3-d grid configuration; linear weighting is used in the cross-field variables ξ and χ . Both weighting methods cause an order $(k\Delta)^2$ flattening of the spectrum at low wavenumbers.

The next step in obtaining the charge density for the Poisson solver is to include the effects due to the metric. The actual charge density is determined from the differential relation

$$\rho = \frac{\partial V'}{\partial V} \sigma = \frac{1}{J} \sigma, \quad (42)$$

where J is given by $J = h_r h_\chi h_\zeta$ and represents the transformation Jacobian for the toroidal coordinate system (h_r includes the effect of the nonuniform grid). This becomes

$$\rho = \frac{\partial \xi / \partial r}{h_\chi h_\zeta} \sigma, \quad (43)$$

where σ is the flat-metric charge density. This operation must be performed in (r, χ) space due to the θ -dependence in h_ζ . The main problem with this approach is when the Jacobian vanishes at some point within the simulation region, as it does in the toroidal system at $r = 0$. This problem can be avoided by placing the boundaries between gridpoints; there will be no grid point at $r = 0$ and thus no singularity. The given method of charge accumulation, together with the filter techniques discussed below, in principle allow the adoption of a wide variety of coordinate metrics. A similar approach to handling the metric was used by Brackbill and Ruppel [19] for two-dimensional particle-in-cell fluid simulations with an adaptive grid.

Finally, the filtering (or finite-size-particle shaping) must be considered. It proves advantageous in this algorithm to filter the pseudo-charge density σ , and not the charge density itself (discussed more fully in the next section). A commonly-used filter in particle simulations is the convolution of the accumulated charge density with a shape-factor, often a Gaussian, which is performed in k-space :

$$\sigma(\mathbf{k}) \rightarrow \sigma(\mathbf{k}) \exp \left(-[\mathbf{k} \cdot \mathbf{a}]^2 / 2 \right),$$

where k is the wavevector and a is a measure of the particle size. This type of global filtering operator has the advantages of flexibility and ease of implementation, and can be used to partially compensate for errors introduced in the Poisson solver or the sampling method [8, Appendix B]. At present, a simple Gaussian shape-factor in the χ and ζ variables is used.

The use of a global filtering operator for a highly nonuniform or non-Cartesian coordinate is questionable, however. For the radial variable in a cylindrical system, a non-standard (Bessel) representation must be used in order to have a rigorously convergent eigenfunction expansion. This problem can be avoided by considering a region far from the origin, as in the work of Cheng and Okuda [4]. In the case of a nonuniform grid, the non-locality of the above operator is somewhat inappropriate, although a transform in the stretched variable ξ is reasonable if a rigorously convergent transform can be found.

For such situations a better alternative exists, known as *digital* (or real-space) filtering [8, Appendix C]. The digital filter employed here is written as \hat{S}_D^N , denoting the N -fold application of the simple filter

$$\hat{S}_D(\sigma_i) = \frac{\sigma_{i-1} + 2\sigma_i + \sigma_{i+1}}{4}, \quad (44)$$

where i stands for the ξ -coordinate. This is known as a *binomial* filter and is equivalent to a gaussian shape factor in the limit of $N \rightarrow \infty$. To see this, we write this filter in terms of its k -space representation in a uniform grid system [8] and then expand, assuming small $k\Delta$:

$$\begin{aligned} \hat{S}_D^N(k) &= \left(\cos \frac{k\Delta}{2} \right)^{2N} \\ &\approx \left[1 - \left(\frac{k\Delta}{2} \right)^2 \right]^N. \end{aligned}$$

When using this filter, the effective particle size will increase as N increases. We let

$$a_* = \sqrt{\frac{N}{2}} \Delta, \quad (45)$$

whereupon the $N \rightarrow \infty$ limit of the filter becomes

$$\lim_{N \rightarrow \infty} \hat{S}_D^N(k) = \lim_{N \rightarrow \infty} \left[1 - \frac{k^2 a_*^2 / 2}{N} \right]^N$$

$$\equiv \exp\left(-\frac{k^2 a_*^2}{2}\right).$$

Although this only holds exactly in the large N , small $k\Delta$ limit, it nevertheless gives an estimate of the effective particle size associated with the application of the filter. The total filtering operator can thus be written as

$$\sigma(\xi, k_\chi, k_\zeta) \rightarrow \hat{S}_D^N(\sigma) \hat{S}(k), \quad (46)$$

where $\hat{S}(k)$ is given by

$$\hat{S}(k) = \exp\left[-(k_\chi^2 a_\chi^2 + k_\zeta^2 a_\zeta^2)/2\right]. \quad (47)$$

The electric field filtering and interpolation is handled similarly, except that no transformation weighting is required. The fundamental algorithm may be summarized as follows :

1. Accumulate & filter pseudo-charge density σ .
2. Obtain charge density from $\rho = \sigma/J$.
3. Solve Poisson equation and determine \mathbf{E} .
4. Filter electric field and interpolate to particle positions.
5. Push particles.

4.3 Momentum and Energy conservation

Here we examine the ideas of the previous section in a more formal manner, and show that momentum and energy is conserved by the algorithm. We find that conservation of momentum (i.e. no numerical self-force) dictates a specific relationship between the distributions used for interpolation/filtering of the charge density and those used for the electric field (being identical only in the slab limit). The presence of non-periodic boundaries ensures the presence of a physical self-force on a particle—via the induced image charge. This term can be accounted for in the conservation laws [20]. A reasonable choice of interpolating/filtering distributions for the electric field leads to the straightforward methods given in the previous section.

To shed light on the questions of momentum and energy conservation, we formally express the field solver steps in a general metric. Starting with the point-particle distribution, one first interpolates to a mesh; this is mathematically equivalent with convolving the point-particle distribution with a given interpolating distribution. Similarly, the filtering process is regarded as a convolution of the resulting function with a given filtering distribution. Each of these processes has the result of spatially spreading the charge, giving rise to the well-known finite-size particle effect [6].

Although they are distinct processes, the interpolation and filtering processes are closely related, and shall initially be represented in terms of some general operator \mathcal{L} . Using a continuous space variable, these operations on a function f are written as

$$\mathcal{L} \circ f(\mathbf{x}) = \int J(\mathbf{x}') d\mathbf{x}' f(\mathbf{x}') \mathcal{L}(\mathbf{x}'; \mathbf{x}), \quad (48)$$

where $\mathcal{L}(\mathbf{x}'; \mathbf{x})$ is the interpolating or filtering distribution, and $J(\mathbf{x}') d\mathbf{x}' = J(\mathbf{x}') dx'_1, \dots, dx'_n$ is the volume element, for dimensionality n . In this operation we are simply summing the weights given to the point at \mathbf{x} from every point in space (\mathbf{x}'); this will naturally depend on the geometry at \mathbf{x}' . Note that in general $\mathcal{L}(\mathbf{x}'; \mathbf{x}) \neq \mathcal{L}(\mathbf{x}; \mathbf{x}')$, where the coordinate in the second position denotes the fixed point.

We adopt interpolating distributions \mathcal{W}^* and \mathcal{W} for acting on the charge density and electric field, respectively, which are applied in the fashion of Eq. (48). Similarly, the filtering distributions are written \mathcal{S}^* and \mathcal{S} . We will prove the following :

Given an interpolating or filtering distribution \mathcal{L} convolved with the electric field according to Eq. (48), and an analogous distribution \mathcal{L}^* convolved with the charge density, then momentum is conserved by the algorithm if

$$\mathcal{L}^*(\mathbf{x}; \mathbf{x}') = \mathcal{L}(\mathbf{x}'; \mathbf{x}). \quad (49)$$

We take the charge density distribution $\rho(\mathbf{x}')$ to given by that of an individual point source at \mathbf{x}_j :

$$\begin{aligned} \rho_j(\mathbf{x}') &= q_j \delta(\mathbf{x}' - \mathbf{x}_j) \\ &\equiv \frac{q_j}{J(\mathbf{x}')} \prod_{i=1}^n \delta(x'_i - x_j). \end{aligned} \quad (50)$$

The calculation of the self-field of an individual particle proceeds as follows :

1. Interpolation. The charge density for a single particle of charge q_j and position \mathbf{x}_j is given by

$$\begin{aligned}\tilde{\rho}_j(\mathbf{x}) &= \int J(\mathbf{x}') d\mathbf{x}' \rho_j(\mathbf{x}') \mathcal{W}^*(\mathbf{x}'; \mathbf{x}) \\ &= q_j \mathcal{W}^*(\mathbf{x}_j; \mathbf{x}).\end{aligned}\tag{51}$$

2. Filtering. The charge density is filtered to obtain the source term in the field equations,

$$s_j(\mathbf{x}) = \int J(\mathbf{x}') d\mathbf{x}' \tilde{\rho}_j(\mathbf{x}') \mathcal{S}^*(\mathbf{x}'; \mathbf{x}).\tag{52}$$

3. Field solution. Here we are solving for \mathbf{E} in terms of s_j :

$$\nabla \cdot \mathbf{E} = 4\pi s_j(\mathbf{x}).\tag{53}$$

4. Filtering. The filtered electric fields are given by

$$\mathbf{e}_j(\mathbf{x}) = \int J(\mathbf{x}') d\mathbf{x}' \mathbf{E}_j(\mathbf{x}') \mathcal{S}(\mathbf{x}'; \mathbf{x}).\tag{54}$$

5. Interpolation to particles. The force on the j th particle (due to its own field) is

$$\mathbf{F}_j(\mathbf{x}_j) = \int J(\mathbf{x}) d\mathbf{x} \mathbf{e}_j(\mathbf{x}) q_j \mathcal{W}(\mathbf{x}; \mathbf{x}_j).\tag{55}$$

Combining Eqs. (54,55), the force on the j th particle due to its self-field (written as \mathbf{F}_{jj}) becomes

$$\mathbf{F}_{jj} = \int J(\mathbf{x}') d\mathbf{x}' \mathbf{E}_j(\mathbf{x}') \int J(\mathbf{x}) d\mathbf{x} q_j \mathcal{W}(\mathbf{x}; \mathbf{x}_j) \mathcal{S}(\mathbf{x}'; \mathbf{x}).\tag{56}$$

The source term to the Poisson solver is obtained by combining Eqs. (51,52), giving

$$s_j(\mathbf{x}') = \int J(\mathbf{x}) d\mathbf{x} q_j \mathcal{W}^*(\mathbf{x}_j; \mathbf{x}) \mathcal{S}^*(\mathbf{x}; \mathbf{x}'),\tag{57}$$

where the integration variable has been changed to \mathbf{x} . We now write the conservation of momentum relation given by Decyk [20] for a particle in its own field as

$$\frac{d\mathbf{P}_{jj}}{dt} - \oint_S \mathbf{T}_{jj} \cdot d\mathbf{S} = 0, \quad \frac{d\mathbf{P}_{jj}}{dt} \equiv \mathbf{F}_{jj}, \quad (58)$$

where \mathbf{T} is the maxwell stress tensor in the electrostatic limit :

$$\mathbf{T} = \frac{1}{4\pi} \left[\mathbf{E}\mathbf{E} - \frac{1}{2}(\mathbf{E} \cdot \mathbf{E}) \mathbf{I} \right]$$

and \mathbf{I} is the identity tensor (note $d\mathbf{P}/dt = \partial\mathbf{P}/\partial t + \nabla \cdot \mathbf{M}$ in Decyk's notation). Enforcing the electrostatic condition ($\nabla \times \mathbf{E} = 0$), the conservation law takes on the form

$$\begin{aligned} \mathbf{F}_{jj} &= \int dV \nabla \cdot \mathbf{T}_{jj} \\ &= \int dV \mathbf{E}_j (1/4\pi) \nabla \cdot \mathbf{E}_j. \end{aligned} \quad (59)$$

Comparison of Eqs. (56), (57), and (59) leads immediately to the requirement that

$$\mathcal{W}(\mathbf{x}; \mathbf{x}_j) \mathcal{S}(\mathbf{x}'; \mathbf{x}) = \mathcal{W}^*(\mathbf{x}_j; \mathbf{x}) \mathcal{S}^*(\mathbf{x}; \mathbf{x}') \quad (60)$$

for momentum to be conserved in a general metric. In practice, however, it is advantageous to independently require

$$\mathcal{L}(\mathbf{x}; \mathbf{x}') = \mathcal{L}^*(\mathbf{x}'; \mathbf{x}) \quad (61)$$

for each interpolating or filtering distribution (i.e. \mathcal{W} or \mathcal{S}), clearly satisfying Eq. (60).

Conservation of energy now follows directly. For the given particle, we have $dW_k/dt = v_j \cdot F_{jj}$, or

$$\frac{dW_k}{dt} = \int J(\mathbf{x}') d\mathbf{x}' \mathbf{j}(\mathbf{x}') \cdot \mathbf{E}_j(\mathbf{x}'), \quad (62)$$

where \mathbf{j} denotes the current density. This leads immediately to the energy conservation law given in Ref. [20]. We note that this treatment ignores possible errors in the field solver and the finite time-step; in an actual simulation neither energy nor momentum is exactly conserved. However, these are relatively small variations and serve as a check on the integrity of the simulation.

The arguments given here show that this viewpoint is not modified by the adoption of a complex metric in our treatment.

It remains to further specify the form of the interpolating and filtering distributions. We shall require that the operator \mathcal{L} (acting on the electric field) be *neutral*, so that the distribution satisfies

$$\int J(\mathbf{x}') d\mathbf{x}' \mathcal{L}(\mathbf{x}'; \mathbf{x}) = 1. \quad (63)$$

For this to hold independent of metric, the distribution must transform like a density :

$$\mathcal{L}(\mathbf{x}'; \mathbf{x}) = \frac{1}{J(\mathbf{x}')} \prod_{i=1}^n L_i(x'_i - x_i) \quad (64)$$

$$\equiv \frac{1}{J(\mathbf{x}')} L(\mathbf{x}' - \mathbf{x}), \quad (65)$$

where $L_i(x'_i - x_i)$ is the flat-metric distribution of the i th coordinate, and $L(\mathbf{x}' - \mathbf{x})$ denotes the total flat-metric distribution (i.e. the product of the individual coordinate distributions). Note that $L_i(x'_i - x_i) = L_i(x_i - x'_i)$. Writing flat-metric distributions W and S analogously, the electric field filtering operation (Eq. (54)) reduces to

$$\mathbf{e}_j(\mathbf{x}) = \int d\mathbf{x}' \mathbf{E}_j(\mathbf{x}') S(\mathbf{x}' - \mathbf{x}) \quad (66)$$

and electric field interpolation (Eq. (55)) reduces to

$$\mathbf{F}_j(\mathbf{x}_j) = \int d\mathbf{x} \mathbf{e}_j(\mathbf{x}) q_j W(\mathbf{x} - \mathbf{x}_j). \quad (67)$$

A simple, metric-independent result thereby emerges. Any of the standard flat-metric distributions for interpolation or filtering (such as described in the previous section) may be used in place of L .

The conservation of momentum requirement (Eq. (61)) leads to charge density interpolation and filtering distributions of the form :

$$\mathcal{L}^*(\mathbf{x}'; \mathbf{x}) = \frac{1}{J(\mathbf{x})} L(\mathbf{x}' - \mathbf{x}), \quad (68)$$

where the flat-metric distribution L is identical to that used for the electric field. The only difference here, though an important one, is the presence of

the $1/J(\mathbf{x})$ term before the flat-metric distribution instead of $1/J(\mathbf{x}')$. The charge density accumulation (Eq. (51)) reduces to

$$\tilde{\rho}_j(\mathbf{x}) = \frac{\sigma_j(\mathbf{x})}{J(\mathbf{x})}, \quad (69)$$

where σ is the flat-metric charge density discussed in the previous section. The charge density filtering (Eq. (51)) is given by

$$s_j(\mathbf{x}) = \frac{1}{J(\mathbf{x})} \int d\mathbf{x}' \sigma_j(\mathbf{x}') S(\mathbf{x}' - \mathbf{x}). \quad (70)$$

Thus we see that using the non-neutral distributions \mathcal{W}^* and \mathcal{S}^* in the charge density calculation is equivalent to first accumulating and filtering the flat-metric density σ , then dividing by the transformation weight $J(\mathbf{x})$. As remarked previously, this treatment avoids problems in the toroidal system at $r = 0$ from the vanishing of the Jacobian, if no gridpoint is there. Both the interpolator and filter conserve charge if permitted by the boundary conditions (see Appendix B for discussion of charge conservation by the filter).

By contrast, one might reverse the treatment adopted here, using the non-neutral distributions for interpolation/filtering of the electric field. The interpolated electric field in this case resembles the “flux-weighted” average described in Ref. [8], with the charge accumulation leading to

$$\rho_j(\mathbf{x}) = \frac{q_j}{J(\mathbf{x}_j)} W(\mathbf{x}_j - \mathbf{x}). \quad (71)$$

The presence of $1/J(\mathbf{x}_j)$ in the above expression causes difficulties if the particle comes too close to $r = 0$. This choice also causes problems with charge conservation by both the interpolator and the filter. Nevertheless, there are some advantages to either approach; the simplicity and flexibility of Eqs. (66,67) for the electric fields and Eqs. (69,70) for the charge density proved to be the overriding factor in our choice.

4.4 Boundary handling

In a simulation code, the boundary conditions enter in several places—the field solver, the filtering of the field quantities, and the interpolation between

the particles and the grid. For the particle/grid interpolations, the boundary conditions can be handled in a number of ways: decision-making (IF-THEN constructs), indirect lookup-table methods, or guard cells. The guard cell method used in this research employs an extra grid-point immediately outside the system that is mapped back on to the interior values in a manner consistent with the boundary conditions. This has the advantage of speed—the interpolation routines do not have to deal explicitly with the boundaries, nor are there any indirect lookup tables that may slow execution or inhibit vectorization. Another advantage of the guard cell technique is that unusual boundary conditions can be easily handled, since the boundary treatment (filling and emptying guard cells) is completely isolated from the rest of the calculation. In particular, the transition to the $r = 0$ boundary conditions at the inner boundary is eased by a guard cell formulation.

The boundary conditions at $r = 0$ must ensure continuity there, and the conditions for single-valuedness and continuity of first derivative are given by Aydemir and Barnes [21]. For a scalar field $f(r, \theta, \phi)$, we require that

$$\begin{aligned} \left. \frac{\partial f}{\partial r} \right|_{r=0} &= 0 & \text{for } m = 0 \\ f(r, m, n)|_{r=0} &= 0 & \text{for } m \neq 0, \end{aligned}$$

where m and n are the mode numbers for the transformed function f . A vector field $u(r, \theta, \phi)$ must satisfy the conditions

$$\begin{aligned} u_r(r, m, n)|_{r=0} + im u_\theta(r, m, n)|_{r=0} &= 0 & \text{for } |m| = 1 \\ u_r(r, m, n)|_{r=0} &= 0 \\ u_\theta(r, m, n)|_{r=0} &= 0 & \text{for } |m| \neq 1, \end{aligned}$$

for the (r, θ) components; the u_ϕ component has the same boundary conditions as a scalar function. The boundary conditions for a scalar field apply to the charge density, potential, and E_z field component; those for a vector field apply to the E_r, E_χ field components.

The implementation of the continuity boundary conditions is straightforward when the k -space (in θ) values of the field are readily available, as with the electric fields. The boundary handling of the pseudo-charge density σ entails mapping the guard cell particle contributions to the interior of the σ array using the continuity boundary conditions. It may in some cases be

preferable to perform the boundary handling in real space. This is accomplished by redistributing the guard cell values among the first grid points, so that the scalar field continuity condition is satisfied. A particle inside the "central cell" can be linearly interpolated between the first radial grid point and the virtual point at the origin; the contribution at the origin is then uniformly spread out among all the neighboring theta points. Although the interpolation has already been done, this numerical procedure is possible because all the guard cell values have been kept. Thus a particle exactly at $r = 0$ will assign charge equally to all of the points on the boundary of the central cell, a result that is not only aesthetically appealing but compatible with the continuity boundary condition there.

5 Particle pusher

5.1 Ions

The usual scheme for the time advancement of the ions in particle simulations employs the leap-frog method [17]. Applied to the system of differential equations employed in this model, the leap-frog method is non-dissipative and second-order accurate in the time-step. Application of this method to a general non-cartesian coordinate system is difficult, however. We start with the equations of motion for an arbitrary metric :

$$h_k \ddot{q}_k + \sum_j h_j \dot{q}_j \left(2\dot{q}_k \frac{\partial h_k}{h_j \partial q_j} - \dot{q}_j \frac{\partial h_j}{h_k \partial q_k} \right) = \frac{q}{m} \left\{ E_k + \sum_{ij} h_i \dot{q}_i B_j \epsilon_{ijk} \right\}, \quad (72)$$

where a dot refers to a temporal differentiation. Equation (72) can be directly derived from the Lagrange equations, using the transformed Lorentz force law. A simpler representation is obtained in terms of the angular momenta, giving $q_k = l_k / h_k^2$ for the coordinates, and

$$\frac{\dot{l}_k}{h_k} - \sum_j \left(\frac{l_j}{h_j} \right) \left(\frac{l_j}{h_j^2} \frac{\partial h_j}{h_k \partial q_k} \right) = \frac{q}{m} \left\{ E_k + \sum_{ij} \frac{l_i}{h_i} B_j \epsilon_{ijk} \right\}, \quad (73)$$

for the angular momenta. If the toroidal metric is used, the orbit equations become

$$\begin{aligned} \ddot{u} - \kappa_\chi \frac{l_\chi^2}{(1 + \kappa_\chi u)^3} - \epsilon_\zeta \frac{l_\zeta^2 \cos \theta}{(1 + \epsilon_\zeta (1 + \kappa_\chi u) \cos \theta)^3} &= F_u \\ \frac{\dot{l}_\chi}{(1 + \kappa_\chi u)} + \epsilon_\zeta \frac{l_\zeta^2 \sin \theta}{(1 + \epsilon_\zeta (1 + \kappa_\chi u) \cos \theta)^3} &= F_\chi \\ \frac{\dot{l}_\zeta}{(1 + \epsilon_\zeta (1 + \kappa_\chi u) \cos \theta)} &= F_\zeta, \end{aligned} \quad (74)$$

in which the right-hand-terms are determined from Eq. (73). One standard way of handling the rotation of the velocity vector in slab geometry is to finite difference the velocities using

$$\dot{v}_k^n = \frac{v_k^{n+1/2} - v_k^{n-1/2}}{\delta t} + \mathcal{O}(\delta t^2) \quad (75)$$

$$v_k^n = \frac{v_k^{n+1/2} + v_k^{n-1/2}}{2} + \mathcal{O}(\delta t^2), \quad (76)$$

which give a centered, second-order accurate finite-difference equation. The components are then collected into a matrix equation

$$\mathbf{M}\mathbf{v}^{n+1/2} = \mathbf{N}\mathbf{v}^{n-1/2} + \frac{q}{m} \delta t \mathbf{E}, \quad (77)$$

where \mathbf{M} , \mathbf{N} , and \mathbf{E} all depend on quantities at the n th time level. This equation is inverted analytically to give the velocities at the $n + \frac{1}{2}$ timestep in terms of those at the $n - \frac{1}{2}$ timestep.

A straightforward implementation of this method will not work in a general coordinate system due to the nonlinear terms in the equations of motion (72)–(73). In this case there is a choice of three quantities that may be considered given at the half-integer timestep and correspondingly finite-differenced. These are the time derivative of the coordinates \dot{q}_k , the velocities $h_k \dot{q}_k$, or the angular momenta $h_k^2 \dot{q}_k$. Regardless of which variable is differenced, the nonlinearities remain, causing the standard leap-frog method to fail.

One alternative is to adopt a predictor-corrector type approach based on the smallness of the nonlinear terms in equations (72) or (73). The problem with this method is that the second term in Eq. (74) is not small and is a poor expansion parameter in the typical toroidal case. The resulting decenteredness causes the rapid cyclotron motion to decay [22]. Although the damping of cyclotron motion is often an intended effect in an implicit code [23], it is undesirable for it be an unavoidable by-product of the metric if the full ion dynamics are to be retained. However, if gyroradius effects are unimportant, we can utilize this scheme by starting with the gyroenergy being zero (in this case, we might deliberately decenter Eq. (74)).

The solution to this problem is to push in a coordinate system that is better-suited to the physics of the gyrating particles. The approach chosen here is simple—to push completely in Cartesian coordinates. This method, though involving time-consuming transformations, has several advantages. First, it is rigorously centered and second-order, with little or no numerical damping. A test run comparing to the previous method showed excellent energy conservation over the course of the run. The second advantage of the transformation approach is that it can be used for any coordinate system, as long as the transformations can be done accurately. (More precisely, there must exist a transform function together with an exact inverse—since the velocity components never “leave” Cartesian coordinates, the rotation will be handled correctly, even if the transform itself is not exact). For the toroidal

system the transformations are straightforward but somewhat expensive; it is likely, however, that the cost of a sophisticated predictor-corrector routine as previously given is comparable (although no detailed timing comparisons were performed).

The only remaining concern is the amount of numerical diffusion associated with this method of pushing. The amount of deviation from the field line for a single particle in one timestep is proportional to $(v_{\parallel} \delta t)^2$ since the pushing method is second order accurate. This quantity will be small for the ion species in most cases. For a Lorentz-force pusher, the need to accurately reproduce the cyclotron motion is the dominant constraint; numerical diffusion will not be an issue except when large anisotropies in the ion temperature distribution exist ($T_{\parallel} \gg T_{\perp}$). Test runs of this pusher with an ion temperature gradient (no electric fields, otherwise typical simulation parameters) have shown negligible numerical diffusion over long time scales. In addition, single particle checks (see Sec. 6.1) show excellent conservation of particle invariants for the zero electric field case.

5.2 Electrons

In a strongly magnetized plasma, a standard technique to follow electron dynamics involves the adoption of a guiding center formulation of the equations of motion. An algorithm for this method in particle simulation was first given by Cheng and Okuda [4], and has been employed in many recent simulations [24,25].

In toroidal geometry, the guiding-center drift equations include the important first order Larmor radius magnetic effects—mirroring (banana orbits), drifts from the flux surface, and so on. The primary elements missing in this description are the polarization current and the finite Larmor radius electric effects; these are usually small for electrons in a strongly magnetized plasma ($\rho_e \lesssim \lambda_e$). For non-relativistic dynamics ($E \ll B$) and low frequencies the electron drift equations given by Northrop [26] reduce to

$$\mathbf{v}_d = \mathbf{u}_e + \frac{\mathbf{b}}{\Omega_e} \times \left\{ (\mu/m) \nabla B + v_{\parallel}^2 (\mathbf{b} \cdot \nabla \mathbf{b}) \right\} \quad (78)$$

$$\frac{dv_{\parallel}}{dt} = -\frac{e}{m} E_{\parallel} - \frac{\mu}{m} \mathbf{b} \cdot \nabla B, \quad (79)$$

where $\mathbf{u}_e = \mathbf{E} \times \mathbf{b}/B$, $\mathbf{b} = \mathbf{B}/B$, $\mu = \frac{1}{2} m v_{\perp}^2 / B$, $\Omega_e = eB/mc$, and m is the

electron mass. The particles would normally be pushed by time differencing

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}_d + v_{\parallel} \mathbf{b}. \quad (80)$$

However, it is seen from Eq. (78) that the perpendicular drift motion is not compatible with the leapfrog method since the fields and drift velocity are given at the same time level. This is the rationale behind the predictor-corrector method [4] for the perpendicular component, which is third order accurate in δt (and is slightly dissipative, unlike the leap-frog method). The parallel velocity component is pushed in the normal way :

$$v_{\parallel}^{n+1/2} = v_{\parallel}^{n-1/2} + F_{\parallel}^n \delta t. \quad (81)$$

The prediction step uses the velocities at time step n and the positions at time step $n - 1$ to estimate the positions at time step $n + 1$:

$$(\mathbf{x}^{n+1})^* = \mathbf{x}^{n-1} + 2\delta t \left(\mathbf{v}_d^n + \frac{1}{2} [v_{\parallel}^{n+1/2} + v_{\parallel}^{n-1/2}] \mathbf{b} \right). \quad (82)$$

The predicted positions are used to calculate values for the fields at the next time-step, giving the drift velocity for the correction step :

$$\begin{aligned} (\mathbf{v}_d^{n+1})^* &\equiv \mathbf{v}_d^{n+1} [(\mathbf{x}^{n+1})^*] \\ \mathbf{x}^{n+1} &= \mathbf{x}^n + \delta t \left(\mathbf{v}_{\parallel}^{n+1/2} + \frac{1}{2} [\mathbf{v}_d^n + (\mathbf{v}_d^{n+1})^*] \right). \end{aligned} \quad (83)$$

It is important to properly time-center the magnetic field components in this equation since inhomogeneities may be strong. We currently use only magnetic fields that can be expressed analytically, so that the ∇B and $\mathbf{b} \cdot \nabla \mathbf{b}$ terms are known exactly, avoiding the need for magnetic field interpolation. The $\mathbf{b} \cdot \nabla \mathbf{b}$ term is evaluated using the identity $\mathbf{b} \cdot \nabla \mathbf{b} = [\nabla B - \mathbf{b} \times (\nabla \times \mathbf{B})]/B$. Derivation of the necessary drift terms is then straightforward, given expressions for the gradient and curl in an arbitrary coordinate system [27].

The only remaining subtlety is in the advancement of the particle coordinates from the velocities. For an arbitrary metric, the velocity for the j th component is $h_j \dot{q}_j$, so that Eqs. (82)–(83) become

$$\begin{aligned} (q_j^{n+1})^* &= q_j^{n-1} + 2\delta t v_j^n / h_j^n & (\text{predict}) \\ q_j^{n+1} &= q_j^n + \delta t v_j^{n+1/2} / h_j^{n+1/2} & (\text{correct}), \end{aligned}$$

where h_j is in general a function of all three coordinates, (q_1, q_2, q_3) . The prediction step utilizes the metric components h_j^n , obtained from the coordinates at the n th time level. The correction step, however, requires unknown quantities $h_j^{n+1/2}$. One way of addressing this problem is to use the predicted positions in the calculation of h_j :

$$\begin{aligned} h_j^{n+1/2} &\equiv h_j(\mathbf{q}^{n+1/2}) \\ &\approx h_j([\mathbf{q}^n + (\mathbf{q}^{n+1})^*] / 2). \end{aligned} \quad (84)$$

A more accurate method also exists when the metric is of the form $h_1 \equiv \text{constant}$, $h_2 \equiv h_2(q_1)$, $h_3 \equiv h_3(q_1, q_2)$, as in the toroidal system. This allows the push to be ordered so that the coordinates needed for each metric component are known (to second order in δt). The results of the r -push enable the calculation of h_χ for the χ -push, which enables the calculation of h_ζ for the ζ -push. However, the accuracy gained by using this ordering does not change the overall accuracy of the push significantly, as seen in long test runs. Therefore, we use the predicted coordinates as in Eq. (84) for the push, to allow simple application of the pusher algorithm in more general metrics.

A major liability of this predictor-corrector algorithm is the need to repeat the entire field calculation for the electrons at the predicted positions. This is common to both the present implementation and the usual slab implementation [4], and is costly both in terms of computation and programming effort. The amount of programming effort can be reduced through the use of a modular field solver—the field loop can be reduced to a few subroutine calls within the guiding center pusher routine. Another liability is the increased number of particle quantities necessary. A three-dimensional slab implementation of the guiding center pusher requires nine particle quantities in general—the three coordinates and velocities, and a temporary vector that holds the coordinates of the previous time-step for the predictive step, as well as the predicted positions for the corrective step. The implementation in the toroidal system adds two additional particle quantities—the magnetic moment (μ), which does not evolve but is different for each electron (determined by the initial loading), and the velocity component parallel to the magnetic field ($v_{||}$). The three-component velocity vector holds the projection of the drift velocity onto the (u, χ, ζ) coordinate system.

The guiding center formulation works well when the electrons are “tied” to the field lines, as in tokamak-type magnetic fields, with only small drifts

perpendicular to the field lines. In the time scale of interest in the typical numerical experiment, a typical untrapped electron will execute many orbits about the torus, remaining reasonably localized about a given flux surface. In such a system, the numerical diffusion away from the flux surface caused by the finite time-step is certainly a concern, and is one reason for using a flux coordinate system—the parallel motion causes no numerical diffusion in a Hamada-type coordinate system. (Mercier-type coordinate systems are also free from numerical diffusion perpendicular to the flux surface, though numerical diffusion may occur on the flux surface itself.) For example if we examine the amount of numerical diffusion in a “square torus” configuration (cylindrical coordinate, tokamak fields) due to the poloidal field, we find that the deviation per time step from the flux surface will be on the order of

$$\delta \sim \left(v_{\parallel} \delta t \frac{B_x}{B} \right)^2. \quad (85)$$

Although this is second order in δt , the product $v_{\parallel} \delta t$ for electrons may be appreciable for typical simulation parameters, creating a definite concern about numerical diffusion in this system.

5.3 Particle boundary handling

Finally we consider the effect of a vanishing Jacobian ($r = 0$ in the toroidal system) on the particle pushers. No modification of the ion push is required since the ions are being pushed in a Cartesian coordinate system, but the $1/r$ terms in the electron guiding-center pusher require some special handling. Although the number of electrons near the origin is small, eventually an electron will pass close enough to $r = 0$ to cause numerical difficulties. This problem is resolved by letting electrons undergo a collision when passing too close to the origin. We can imagine a small, hard cylinder (or torus) centered at $r = 0$; the continuity condition demands that an electron passing within the critical radius will exit with its θ coordinate shifted by 180° (i.e. $r \rightarrow -r$). Since the electron orbits in the (r, θ) plane are drift-dominated at the origin, and the dominant magnetic drifts vertical, this treatment is reasonable. The location of this critical radius can be estimated given the maximum tolerable theta increment in a given time-step for a thermal particle, and assuming the drift is dominated by the gradient-B term. The critical radius is then given

approximately by

$$\frac{r_{\min}}{\bar{\Delta}} \sim \frac{4\epsilon_{\zeta}(\omega_e/\Omega_e)v_e^2}{N(\Delta\theta)_{\max}}, \quad (86)$$

where $\bar{\Delta}$ is the average grid spacing, N is the number of radial grid-points, ω_e and Ω_e are the electron plasma and cyclotron frequencies, and v_e is the electron thermal velocity, normalized to $\omega_e\bar{\Delta}$. The maximum tolerable theta increment is of order unity, where the distortion of the “true” particle motion due to the finite integration step rivals that from a collision with the artificial boundary near $r = 0$. Using typical simulation parameters, we obtain

$$\frac{r_{\min}}{\bar{\Delta}} \sim 10^{-2}-10^{-3}.$$

The smallness of this value indicates that a collision is not likely to influence the calculated fields significantly. Depending on the plasma profile and other simulation parameters, an actual run of approximately 50,000 particles suffers at most only a few collisions per time-step at the origin; often no collisions occur.

6 Tests

6.1 Test particle results

In actual plasma devices the magnetic field structure is often complicated and usually highly nonuniform. The resultant magnetic drifts and trapped particles can have a large effect on the stability of the plasma. It is therefore important to include these effects in a simulation in a natural way. In this paper the only restriction we place on the form of the prescribed magnetic fields is that the flux surfaces be nested, concentric tori. This restriction is imposed so that there exist no zeroth-order (in Larmor radius) particle fluxes into the boundaries, and to minimize numerical diffusion. More general magnetic field configurations are possible, while still satisfying this condition, by using a generated grid for storage of the cross-field variables.

The magnetic fields adopted here are essentially tokamak fields—a vacuum field for the toroidal component with the poloidal component specified in terms of the safety factor. These can be written

$$B_\zeta = B_{\zeta_0} \frac{R_0}{R} \quad (87)$$

$$B_x = \frac{r}{R_0 q(r)} B_\zeta, \quad (88)$$

where the safety factor $q(r)$ is usually an increasing function of r . The parallel motion of the particles will be predominantly in the toroidal direction, since r/qR_0 is usually much smaller than unity. However, the poloidal component guarantees that a particle on the outside of the torus will eventually move into a region of stronger magnetic field, causing low parallel-velocity particles to reverse directions. Coupled with the dominant inhomogeneous magnetic field drifts, this gives rise to the well-known banana orbit in this system. Test particle trajectories reproduce this behavior in the toroidal code, both for the full-dynamics ions and the guiding center electrons (see Figure 4). For this test, no electric fields are used, and identical parameters are used for electrons as ions.

The conservation of various particle invariants like energy, magnetic moment (μ), and the toroidal canonical momentum often play central roles in plasma phenomena. Therefore we closely examine the conservation of these quantities by our pusher algorithms in the absence of electric fields. The time

histories of these quantities, for a representative pair of particles, is given in Figures 5–6. We shall refer to a particle as either a Lorentz or a drift particle, depending on the pusher method used. The particle quantities shown in the plots are the parallel velocity v_{\parallel} , kinetic energy E_k , canonical toroidal momentum p_{ϕ} , toroidal angular momentum L_{ϕ} , stream function $\psi = RA_{\phi}$, and zero-order magnetic moment $\mu_0 = v_{\perp}^2/2B$ (Lorentz particle only). The two trajectory plots at top are r versus χ (toroidal cross section), and R versus ζ (view from above).

For the most part, the orbits as given by the two algorithms are in excellent agreement. Part of the discrepancy involves small errors in initialization, since the initial drift particle quantities must represent the “averaged” initial Lorentz particle quantities, which is handled only approximately. The zero-order magnetic moment of the Lorentz particle is on the average well conserved, but oscillates once per cyclotron rotation, indicating that higher order terms in the expansion for the magnetic moment are significant for these parameters. This is supported by better conservation of μ_0 at smaller Larmor radius.

Conservation of the energy and canonical toroidal momentum is excellent for the Lorentz particle, and reasonable for the drift particle. The latter shows a distinct variation of the energy as the particle transits the torus (see Figure 6). This variation is not numerical dissipation, but seems to be a result of drift into a region of higher magnetic field strength. This can happen when a non-vacuum field component is present, in which case the guiding center drift is no longer perfectly perpendicular to ∇B (due to the curvature drift term). Improvement in the conservation of both energy and canonical toroidal momentum by the drift pusher is expected if one moves to a phase-space preserving set of drift equations, such as given by Littlejohn [28].

6.2 Fluctuation spectra for a uniform plasma

The dielectric response function for a uniform equilibrium plasma with drift-kinetic electrons is given by

$$\epsilon(\mathbf{k}, \omega) = 1 + \frac{k_e^2}{k^2} [1 + \zeta_e Z(\zeta_e)] + \frac{k_i^2}{k^2} \left[1 + \sum_n \frac{\omega}{\omega - n\Omega_i} \zeta_{in} Z(\zeta_{in}) \Gamma_n(b_i) \right], \quad (89)$$

where Z is the plasma dispersion function [29]; its arguments in the above equation are given by

$$\zeta_e \equiv \frac{\omega}{|k_{\parallel}|(2T_e/m_e)^{1/2}} \quad (90)$$

$$\zeta_{i_n} \equiv \frac{\omega - n\Omega_i}{|k_{\parallel}|(2T_i/m_i)^{1/2}}. \quad (91)$$

Here we have made the assignment $k_{\sigma} \equiv 1/\lambda_{\sigma}$, and introduced the function

$$\Gamma_n(b) \equiv I_n(b) \exp(-b), \quad (92)$$

where I_n is the modified Bessel function.

At this point the usual practice is to look for solutions to Eq. (89) in a frequency regime where the arguments to the plasma dispersion function are either very large or very small, using an asymptotic form of Z to get a tractable expression. A normal mode solution is found by assuming a frequency dependence of the form $\omega = \omega_0 + i\gamma_0$, with $|\gamma_0| \ll |\omega_0|$. For the plasma wave, the relevant regime is [30]

$$k_{\parallel}v_i \ll k_{\parallel}v_e \ll \omega \ll \Omega_i, \quad (93)$$

giving for the real part of the frequency

$$\omega_0^2 \simeq \frac{k_{\parallel}^2 v_e^2}{k^2 \lambda_e^2 + k_{\perp}^2 \rho_s^2}, \quad (94)$$

where $\Gamma(b)$ has been expanded as $\Gamma(b) \simeq 1 - b$ for small b , and $\rho_s^2 \equiv \tau \rho_i^2$, $\tau \equiv T_e/T_i$. If the ion Larmor radius term is neglected, Eq. (94) reduces to $\omega_0 \simeq \omega_e(k_{\parallel}/k)$ which is the usual result.

The above procedure, involving the expansion of both the plasma dispersion function and the exponentially-scaled modified Bessel function, is not well suited for close comparison between theory and simulation. Therefore, we solve Eq. (89) directly, using a complex root finder (IMSL library routine ZANLYT). The full plasma dispersion function is calculated using a continued fraction method. Although it is usual to keep only the $n = 0$ cyclotron harmonic of the ion response term, we find that it is necessary to keep at least the $n = 1$ or $n = 2$ terms for close comparison with the simulation results (for the given parameters).

The tests given here of the simulation plasma in the plasma wave regime consider only a single harmonic in ζ (mode 6) but all of the perpendicular dynamics; this configuration is similar to that studied in association with the toroidal drift mode (discussed in section 7.2). The simulation parameters are given by : $N_p = 50625$ (number of particles), $L_x = L_y = 64 \Delta$, $L_z = 3L_x$, $N_x = N_y = 64$ (slab geometry and uniform grid, with a mode representation in z), $\Delta t = 0.2 \omega_e^{-1}$, $v_{th} = 0.51 \Delta \omega_e^{-1}$, $\tau = 1$, $m_e/m_i = 0.01$, $\Omega_e/\omega_e = 10$, and particle “size” of $a_x \sim a_y = \Delta$ (using two passes of the binomial digital filter in x). The energy was conserved within 0.03% for 2000 time steps. In the post-processor, the potential is Fourier transformed to obtain the k_x dependence, and the spectral density obtained via the maximum entropy method. Peaks in the spectral density are located numerically, giving the frequency spectrum as a function of wavevector.

We plot frequency versus the perpendicular wavevector compared to the theoretical result in Figure 7. In this figure, the solid line represents the theoretical plasma response with the filtering effects included (finite size particle effects). The filtering effects are included in the theory in a very simple way—we replace ω_c^2 by $\omega_c^2 S^2$ in Eq. (89), where $S(\mathbf{k} \cdot \mathbf{a})$ is a gaussian shape factor and \mathbf{a} is the effective particle size. Agreement between the simulation and theory is good; the remaining systematic error is likely due to grid effects, which have not been corrected for in the rootfinder (i.e. modification of the spectrum from the interpolation and Poisson solver). The flattening seen in the theoretical solution at higher values of $k_\perp \rho_i$ is caused by connection to the $n = 2$ ion Bernstein branch.

For fixed k_\perp and k_\parallel , the frequency response is seen to be fairly insensitive to the direction of propagation [22] for the wavenumber range shown in Figure 7. This is significant in light of the treatment of (x, y) in the field solver—finite-difference and digital filter in x , compared to a more usual k -space treatment in y . Further, adoption of a nonuniform grid in x does not change the plasma response significantly (as expected); the observed mode frequency is close to that of the uniform grid case for the given wavenumber range [22]. This confirms the ability of the nonuniform grid to correctly represent long wavelength modes.

An additional mode often examined in magnetized simulation plasmas is the electron or ion Bernstein mode, which is characterized by frequencies near multiples of the cyclotron frequency for each species. In this algorithm, only the ion modes are observed, since the electron cyclotron motion has

been gyro-averaged via the guiding center equations. The direct solution of the plasma response function (Eq. (89)) for the Bernstein harmonics is an ill-conditioned problem, however. In this case we must rearrange the terms in the response function, based on our knowledge that the roots are near the cyclotron harmonics. In particular, we assume $\omega \simeq n\Omega_i$, whereby the ordering of the terms casts the dispersion relation in the form

$$\omega - \frac{n\Omega_i}{1 - \beta(\mathbf{k}, \omega)} = 0, \quad (95)$$

where $\beta(\mathbf{k}, \omega)$ will be small and is given by

$$\beta(\mathbf{k}, \omega) = \frac{k_i^2}{k^2} \zeta_{in} Z(\zeta_{in}) \Gamma_n(b_i) \frac{1}{R(\mathbf{k}, \omega)}. \quad (96)$$

and

$$R(\mathbf{k}, \omega) = 1 + \frac{k_e^2}{k^2} [1 + \zeta_e Z(\zeta_e)] + \frac{k_i^2}{k^2} \left[1 - \sum_{j \neq n} \frac{\omega}{\omega - j\Omega_i} \zeta_{ij} Z(\zeta_{ij}) \Gamma_j(b_i) \right], \quad (97)$$

with ζ_e and ζ_{in} as previously defined.

The complex rootfinder will converge with the dispersion equation in the form given by Eqs. (95)–(97), but we can simplify further for the purely perpendicular-propagating mode. In this case k_{\parallel} is identically zero, so the large argument expansion of the plasma dispersion function must be used. Expressing the frequency as

$$\omega = n\Omega_i (1 + \Delta_n(\mathbf{k}, \omega)), \quad (98)$$

we have

$$\Delta_n(\mathbf{k}, \omega) = \frac{\beta(\mathbf{k}, \omega)}{1 - \beta(\mathbf{k}, \omega)}, \quad (99)$$

with β given by

$$\beta(\mathbf{k}, \omega) = \frac{k_i^2 \Gamma_n(b_i)}{k^2 + k_i^2 \left(1 - \sum_{j \neq n} \omega / (\omega - j\Omega_i) \Gamma_j(b_i) \right)}, \quad (100)$$

which is then used in the rootfinder calculation.

We may obtain an approximate expression for ω by keeping only the largest term in the sum, given by $j = 0$. If we also assume that β is much less than unity, we obtain

$$\Delta_n(\mathbf{k}) \approx \beta \quad (101)$$

$$\approx \frac{k_i^2 \Gamma_n(b_i)}{k^2 + k_i^2 (1 - \Gamma_0(b_i))}. \quad (102)$$

Note that this differs from the expression for $\Delta_n(\mathbf{k})$ given in Ref. [30]. For the purely perpendicular mode there is no contribution from the electron response, since electrons are tied to the field lines and cannot interact with the wave. Only for a finite angle of propagation will electron screening and Landau damping effects enter. In this case (finite k_{\parallel}) one must solve using the full dispersion function to get an accurate dispersion relation, using Eqs. (95)–(97).

For the ion Bernstein mode test we consider the two-dimensional slab geometry limit of the code. The simulation parameters are given by : $N_p = 50625$ (number of particles), $L_x = L_y = 64 \Delta$, $N_x = N_y = 64$ (slab geometry, uniform grid), $\Delta t = 2.0 \omega_e^{-1}$, $v_{th} = 1.0 \Delta \omega_e^{-1}$, $\tau = 1$, $m_e/m_i = 0.04$, $\Omega_e/\omega_e = 5$, and particle “size” of $a_x \sim a_y = \Delta$ as before. The energy was conserved within 0.5% for 2000 time steps. In Figure 8 we show the frequency response versus perpendicular wavenumber. Here the solid line represents the theoretical response with the gaussian filter correction, while the dashed line is the theoretical response alone. For the parameters chosen in this run the plasma response is concentrated in the lowest Bernstein harmonics.

7 Application

7.1 Establishment of the toroidal equilibrium

The study of collective plasma effects must be preceded by consideration of the equilibrium if the time scales of those effects are much shorter than the operation period of the device or plasma lifetime, which is almost always the case. Since particle simulation utilizes multiple charged species, coupled only through the self-consistent electric (or magnetic) fields, enhanced fluctuation levels by charge separation is a concern. It is well known [31] that inappropriate loading of electron and ion profiles in slab geometry can cause fluctuation levels well in excess of equilibrium levels. This energy is concentrated in the $k_y = 0$ mode (for an x variation of the density profile), and is caused by the large-scale separation of charge perpendicular to the magnetic field. This type of enhanced fluctuation level can easily obscure subtle collective effects of interest in an otherwise stable or marginally unstable plasma. Furthermore, charge separation effects can persist for long times in simulations of the type considered here—the collisionality is low, preventing rapid collisional relaxation, and the magnetic fields are static, so that field line tearing cannot occur.

The driving mechanism for charge separation effects in toroidal geometry is the gradient-B and curvature-B drifts. Electrons and ions experience magnetic drifts of opposite direction, leading to the well-known vertical charge separation in a toroidal device without rotational transform [32]. The inclusion of rotational transform, however, ensures that a particle that drifts away from a flux surface will eventually return to it, as it moves along the poloidal component of the magnetic field. This results in confined trajectories for individual particles, as well as a cancellation of the charge separation effects.

In a particle simulation of the toroidal system, however, the transit time for the typical ion is much longer than other characteristic times of the system (an ion gyration period, for example). The initial particle load, which uses a straightforward extension of the slab procedure, leaves the particles in a highly correlated state, and the effects of the rotational transform are not immediately felt. This leads to large-scale charge separation (primarily due to the ion drifts) and electric field fluctuations far in excess of equilibrium values.

The charge separation seen in the toroidal particle simulation is a result

of the loading of non-stationary particle distributions. In theory a stationary distribution can be obtained by utilizing a function of the constants of motion :

$$f \equiv f(\mathbf{x}, \epsilon, \mu, p_\phi) \quad (103)$$

where \mathbf{x} is the coordinate, ϵ is the energy, μ is the magnetic moment, and

$$p_\phi = mRv_\phi + q\psi(r) \quad (104)$$

is the toroidal canonical momentum ($\psi = RA_\phi$). The theoretical procedure for writing the distribution as a function of r is to expand $f(p_\phi)$ with the assumption that

$$mRv_\phi \ll \psi(r) \quad (105)$$

The problem with this approach is that relation (105) is not valid across the entire plasma profile for typical simulation parameters. What is required is a distribution that generates some desired density profile and is stationary to *all* orders in the ion Larmor radius. Such a constraint may be difficult (if not impossible) to satisfy, so we therefore consider alternate methods for loading the particle distributions.

One alternative is to directly load a distribution in terms of the constants of motion as in (103), using some general function of p_ϕ . This procedure will not allow the exact generation of a desired density profile, but a reasonably close match could be obtained through variation of the function $f(p_\phi)$. In applying this to the phase space of the particles, however, an additional problem arises. The function we are trying to load takes on the form of

$$f(\mathbf{x}, p_\phi) = f(r, \theta, v_\parallel, v_\perp) \quad (106)$$

subject to the constraint that the energy distribution $f(\epsilon)$ be specified (usually a constant temperature Maxwellian, but in general the temperature may be a function of r). The distribution function must approximately give the desired density profile, that is,

$$n_{\text{spec}}(r) \approx n(r) = \left\langle \int d\mathbf{v} f(\mathbf{x}, \mathbf{v}) \right\rangle_{\theta, \phi} \quad (107)$$

In addition the resulting pitch angle dependence (μ/ϵ) and density variation in θ must be reasonable. Note that the ϕ dependence (toroidal coordinate)

is ignorable; the gyrophase angle dependence is usually small and has been neglected in the distribution function.

The generation of random variables corresponding to the distribution given by Eq. (106) subject to the given constraints is a complex problem and is likely to be computationally expensive. A straightforward method would utilize a discretization of the variables $(r, \theta, v_{\parallel}, v_{\perp})$ and would require large amounts of temporary storage for an accurate representation of the distribution. A use of the rejection method [33] to generate the random variables would lessen the memory requirements, but either method is expected to be very costly in cpu time for the large number of particles typically employed. Schemes based on the direct loading of Eq. (106) are still being studied, but a completely different approach has proven feasible.

The use of a distribution function that is not a function of the constants of the motion is responsible for the failure of the "standard" slab load in the toroidal system. This distribution is not stationary and will rapidly evolve. In the absence of electric fields the mechanism by which it evolves is known as phase-mixing (since the particles are non-interacting in this case, so that there is no physical dissipation). Eventually a stationary distribution is reached, which may be used as a starting point for our toroidal particle simulation. Thus we generate a stationary distribution by initializing the particle positions in the standard way, then advancing them in time according to the given magnetic field configuration without the self-consistent electric fields (we shall refer to this as an *equalization* run). Phase-mixing eventually results in quasi-stationary distributions. This argument assumes that no zero-order electric potential is necessary for equilibrium.

For this "equilibrium via phase-mixing" to succeed computationally, we require first an efficient method for pushing the particles (primarily ions) on time scales very long compared with the cyclotron gyration. This requirement is satisfied by utilizing the guiding center equations of motion (ordinarily used for the electrons). These equations are accurate even for large Larmor radii in the absence of electric fields, since the magnetic field usually varies slowly on this scale. The second requirement is to develop a criterion by which we can gauge the closeness to stationarity and thereby terminate the initialization procedure. A simple symmetry argument suffices here. The initial load, as well as the time-averaged particle orbits, are vertically symmetric (i.e. symmetric about the $\theta = 0$ and $\theta = \pi$ half-planes). Therefore, since the time-averaged distribution is vertically symmetric as well, the ensemble-averaged

distribution must also share this symmetry if the system is in statistical equilibrium (ergodic theorem). This requirement of vertical symmetry proves to be a good criterion by which to terminate the equalization run.

We illustrate the initialization procedure by following the temperature evolution for the ion species, as they are pushed in the absence of electric fields. The case studied uses a fairly large Larmor radius ($\rho/a = 0.025$, where a is the minor radius), which emphasizes the non-equilibrium effects since these effects scale as some function of $\kappa\rho$ and ρ/r , where κ is the inverse density scale length. We show in Figure 9(a)–(c) plots of $T_{\parallel}(r, \theta) - \langle T_{\parallel} \rangle$ over a cross section of the plasma at three points in time.

Initially (Figure 9(a)) the temperature shows no global vertical asymmetry aside from velocity space fluctuations. The initial load was such that the total energy and magnetic moment had no global theta dependence. At time $t = 500 \Omega_i^{-1}$ (Figure 9(b)) the temperature displays a strong vertical asymmetry, caused by a global upwards shift of the ion distribution due to the gradient-B drifts. The electric fields that would result from the charge separation are much larger than the thermal equilibrium field level. This situation at best causes an artificial modification of the plasma response, or at worst drives the plasma to the boundary, as in a system without rotational transform.

Finally at the much later time $t = 4000 \Omega_i^{-1}$ (Figure 9(c)) vertical symmetry of the temperature distribution has become well established. Any remaining asymmetry results from either velocity space fluctuations or from the very slowly-moving particles, which take much longer to phase-mix than the bulk of the distribution. Likewise, other moments of the distribution (T_{\perp} , v_{\parallel} , μ_0) are seen to regain vertical symmetry as well.

Another interesting facet of the initialization is the appearance of a large scale structure in poloidal angle of the parallel velocity, as well as a finite value for $\langle v_{\parallel} \rangle$ (initially zero). We give v_{\parallel} as a function of space at times $t = 0$ and $t = 4000 \Omega_i^{-1}$ in Figure 9(d),(e). The observed behavior is again a diamagnetic effect, and is characteristic of a return flow [34]. In fluid theory, the return flow is obtained by enforcing incompressibility to first order in the confinement ordering parameter, ρ/L . If no macroscopic temperature gradients are present the parallel fluid velocity is given by [34] :

$$nV_{\parallel} = -\frac{c}{eB_{\theta 0}}p \left[\frac{d \ln n}{dr} + \frac{e}{T} \frac{d\Phi}{dr} \right] \quad (108)$$

where p is the pressure. For a gaussian density profile ($n(r) \sim \exp(-\kappa_0^2 r^2/2)$), and no electric fields this becomes

$$\frac{V_{\parallel}}{v_t} \simeq \kappa_0^2 \rho q(r) (R_0 + r \cos \theta). \quad (109)$$

where v_t is the thermal velocity, q is the safety factor, and ρ is the Larmor radius. Comparing to Figure 9 (e), we see that the θ -dependent part is well described by Eq. (109). The θ -independent part differs significantly, however, which may be due to the initial load and its effect on the temperature variation. In particular, enforcing no global variation of μ_0 and energy initially results in differing T_{\perp} and T_{\parallel} distributions; these become much closer in appearance during the phase mixing procedure, while the μ_0 distribution develops a strong θ dependence. These observations suggest a number of ways in which the initial load may be improved, but such improvements may not obviate the need for an equalization run, since small errors in the load can lead to relatively large charge separation effects.

The initialization procedure can be summarized as follows. The ion species is pushed via the drift pusher until the vertical symmetry criterion is approximately satisfied. On the order of 2000 time-steps ($\delta t \sim 2 \Omega_i^{-1}$) are typically required. The computational expense for this calculation is reasonable since the electric fields are not being evaluated. At the conclusion of the equalization run, particle quantities are output to disk and then re-read for the actual simulation run, making the necessary transformation from guiding-center coordinates to the full dynamics coordinates. The electron positions (guiding centers) are loaded coincident with the ion positions, which ensures approximate quasineutrality at time $t = 0$ (and is equivalent to the method of Naitou *et al.* [31] in the limit of large number of particles). The electron velocity distribution is loaded in the usual way, which ignores any non-equilibrium effects that may be caused by the geometry. Since the electron drifts are much smaller than the ion drifts, this is usually justifiable.

7.2 Drift waves in toroidal geometry

The density gradient driven drift mode has long attracted the attention of plasma confinement researchers and is suspected to be partly responsible for anomalous transport in tokamaks. Also known as the universal mode, it is unstable for an infinitesimally small density gradient in the absence of

magnetic shear. In the sheared slab model, however, it is stabilized due to the outward convection of energy and ion shear damping [35]. Nevertheless, inclusion of toroidal effects fundamentally alters the character of the mode, leading to the unstable toroidicity-induced (TI) mode [36].

In a toroidal configuration, the magnetic fields have a strong variation with the poloidal coordinate θ , and the Laplacian operator itself has θ -dependent terms. This results in coupling between the drift wave eigenmodes at neighboring mode rational surfaces, which can inhibit the convection of energy. Since it is this convection that is primarily responsible for stabilization of the mode in the slab model, the stability properties of the drift wave are substantially modified.

Simulations of the toroidal plasma with the present code have provided additional information about the character of this mode (both linear and nonlinear) and have demonstrated the efficacy of the code. We summarize some of our findings here; for additional detail see Refs. [22,37].

Parameters in the central region of the simulation plasma are chosen comparable to those used in the theoretical calculation of Ref. [38], for which $\epsilon_n \sim 0.1$ ($\equiv L_n/R_0$) and $\hat{s}(\equiv rq'/q) \sim 1$. The simulation employs $k_\perp \rho_i \sim 0.2$, for which the growth rate of the TI-mode is expected to be a maximum [38]. Runs were performed employing a single toroidal mode number ($n = 9$) or a subset of toroidal mode numbers ($n = 7$ through $n = 11$), as well as a cylindrical control run ($n = 9$). Gaussian density profiles were used, and the safety factor rose from $q \sim 0.6$ at $r = 0$ to $q \sim 4.0$ at $r = a$. The poloidal mode spectrum was truncated to ensure that the wave functions associated with the drift mode were sufficiently isolated from the boundaries. For example, the $m = 5$ through $m = 15$ poloidal modes were retained in the single toroidal harmonic run ($n = 9$).

We observe that the universal modes become unstable in the toroidal plasma, while the cylindrical plasma shows no such instability, suggesting the toroidicity-induced nature of the mode. In the run with multiple toroidal harmonics many growing modes are seen, with growth rates and saturation levels elevated nearly a factor of two over the single- n case.

We examine the mode structure of the potential on long time scales via spectral analysis for the toroidal single- n run. A plot of frequency versus radius for the $m = 8$ and $m = 9$ modes (Fig. 10) shows two dominant modes with frequencies below the diamagnetic frequency (other mode numbers show similar behavior). This result is a consequence of the strong coupling of the

poloidal modes in toroidal geometry, which allows multiple drift harmonics to appear. Comparing the frequency dependence on the radius for neighboring mode numbers, we see a dramatic preference for certain frequency bands (which may vary with position), especially at low frequency. The power (not shown) is concentrated in the two lowest frequency bands, although a third frequency band appears at larger r . The frequency of the lowest harmonic is given roughly by $\omega/\omega^* \sim 0.5$, and the second harmonic by $\omega/\omega^* \sim 0.9$. The theoretical value given in Ref. [38] is $\omega/\omega^* \sim 0.4$, which compares favorably to the observed low frequency band. The cylindrical run shows no band structure in its frequency response, as expected.

The radial structure via an interferogram diagnostic shows peaking of the potential near the mode rational surface for the observed frequencies. The waveforms are typically highly oscillatory in r and overlap an appreciable number of adjacent mode rational surfaces. This behavior is a consequence of the weakly damped or unstable, radially extended character of the toroidal mode, in contrast to the rapid radial decay of the slab geometry drift mode eigenfunction. We observe the width of the interfered potential in the toroidal case to be approximately twice that of the cylindrical run, demonstrating the radially-extended nature of the TI-mode. Note that the finite radial width of the TI-mode eigenfunction is caused by the variation of equilibrium quantities with r (such as the diamagnetic frequency), and from our truncation of the poloidal mode space.

8 Summary and Conclusions

The main goal of the present paper involved the extension of particle simulation methods to nearly arbitrary curvilinear coordinates, and to model of a confined plasma in that system. Kinetic and nonlinear effects play an important role in plasma behavior; thus the capability for studying realistic geometries via particle simulation is greatly desired. The toroidal coordinate system considered here has many features of more complex metrics, including the vanishing of the transformation Jacobian at $r = 0$ and the nonseparable field equations. The algorithm developed in the course of this research thus is virtually metric-independent, and extensions of this work to other coordinate systems are envisioned.

The use of a radial nonuniform grid is a natural construct in a cylindrical or toroidal system, giving increased radial resolution while maintaining nearly uniform particle/cell ratios. Implementation of the nonuniform grid is reasonably straightforward, requiring only a finite-difference (real space) handling of the radial coordinate and the adoption of a real-space or digital filter for smoothing of the interpolated quantities. The association of the curvature of space with the grid (instead of with the particles) in the charge density calculation avoids difficulties with the vanishing of the coordinate metric, as well as being conceptually simple. The resulting accumulation/filter/field-solver/filter/interpolation process is shown to be momentum and energy conserving in the continuum limit, as with the usual slab methods.

The particle equations of motion in the toroidal metric contain nonlinearities that makes integration considerably more difficult than in slab geometry. Therefore the ions are advanced using a transformation pusher, which models the cyclotron gyration well. The electron dynamics is modeled via a guiding center formulation, which is well-suited to this system. The parallel motion is separated from the perpendicular drift motion in the usual predictor-corrector scheme, employing only two additional particle quantities than the slab model—the magnetic moment (μ) and velocity along the field line (v_{\parallel}). These pushing algorithms have been extensively tested in a variety of configurations, including tokamak magnetic fields; the particle orbits are consistently obtained regardless of the pushing method with good conservation of particle invariants (energy, magnetic moment, and toroidal canonical momentum).

The ability of the simulation code to reproduce collective plasma effects

has been demonstrated in the code's slab limit, for the plasma wave and ion Bernstein wave. In each case the simulation results closely followed the theoretical response with the exception of a small systematic deviation increasing with mode number (this resulted from grid effects, which was not accounted for in the analysis). A good test is possible only when the theory is well understood, so that the source of discrepancies can be identified. Thus, reliance on tests in the slab limit for testing of the code is reasonable. (Note that the accurate reproduction of the theoretical perpendicular plasma wave response in the slab limit required that the $n = 1$ and $n = 2$ ion cyclotron contributions be kept. This is surprising in light of the "usual" approximations, and points out the importance of verifying the approximations made in the theory when a close comparison and stringent test are sought.) The correctness of the toroidal field-solver itself has been verified via internal consistency checks and through observation of test particle motion.

An additional question which appears with the simulation model concerns the initialization of stationary particle distributions. This is necessary in a particle simulation so that subtle stability effects can be observed. In the toroidal system the broken symmetry in the poloidal variable causes large-scale charge separation when the distributions are loaded in the usual way. This problem has been solved through a "phase-mixing" procedure, for which the ion distribution is allowed to evolve to a steady state (without electric fields), prior to starting the simulation. A quiet simulation thereby results with fluctuation levels on the order of thermal equilibrium levels.

The success of the particle simulation model in modeling the toroidal drift wave is highly significant, for this mode appears only when toroidal effects are considered. This work demonstrates that particle simulation in complex geometries can be an important tool in the understanding of the underlying physics. However, despite the difficulty of the associated theory and the complexity of the observed phenomena, this investigation is only the starting point. The extension to longer time scales via gyrokinetic and/or implicit formulation of the field equations is under consideration, as well as inclusion of self-consistent magnetic field effects. Finally, the methods developed in the course of this work may be well-suited for the geometries encountered in astrophysical problems as well. This might include simulation of the magnetosphere, or the region near a black hole (for which a Schwarzschild metric would be necessary). The general relativistic metric involves the four dimensional nonflat metric, and treatment of this problem has been initiated.

Acknowledgements

One of the authors would like to thank Dr. Jean-Noel Leboeuf for many interesting discussions on the subtleties of particle simulation. The simulation runs described in this paper were performed on the NMFEC computers in Livermore, California. This work was funded by the U.S. Department of Energy Contract No. DE-FG05-80ET53088 and National Science Foundation Contract No. ATM-88-11128.

A Solution of the Poisson equation in a more general metric.

In a general two-dimensional orthogonal metric with coordinates (ξ, η) , the Poisson equation is given by

$$\frac{1}{h_\xi h_\eta} \left[\frac{\partial}{\partial \xi} \left(\frac{h_\eta}{h_\xi} \frac{\partial}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left(\frac{h_\xi}{h_\eta} \frac{\partial}{\partial \eta} \right) \right] \Phi(\xi, \eta) = s(\xi, \eta), \quad (110)$$

where $h_\xi(\xi, \eta)$ and $h_\eta(\xi, \eta)$ are the metric coefficients for ξ and η , respectively. We consider the case of a coordinate system for which the magnetic flux surfaces lie on surfaces of constant radial coordinate (the “magnetic field” in this discussion will mean the background magnetic field only; i.e. that part not generated self-consistently in the simulation). Further, the magnetic field lines must have no sinks or sources in the region of interest (separatrices are not allowed). These assumptions are reasonable for many numerical calculations.

These coordinates are similar to the boundary-fitted coordinates used in fluid simulations, except that the “boundary” (i.e. magnetic flux surfaces) is present everywhere in space. Aside from an arbitrary scale factor $f_j(q_j)$ in each coordinate, these are essentially Mercier coordinates [39]. The construction of the coordinate system might proceed from the solution of the Grad-Shafranov equilibrium equation, yielding the spatial dependence of the poloidal flux. This would serve to define our radial variable, though a stretching function could be used to obtain the desired radial grid spacing. An orthogonalization procedure (e.g. see Ref. [17]) gives the spatial dependence of the poloidal coordinate. For an axisymmetric magnetic field structure, toroidal angle ϕ may be used as the toroidal coordinate.

Iterative methods of solving Eq. (110) via a rapid elliptic solver have been discussed in the literature (see Ref. [40]); these have typically considered a much more general case. With the assumptions given above, however, the problem becomes much simpler. It was shown by Potter and Tuttle [17] that for the stated conditions the metric components must satisfy

$$h_\xi(\xi, \eta)/h_\eta(\xi, \eta) = f_1(\xi)/f_2(\eta),$$

whereupon Eq. (110) becomes

$$\left[\frac{1}{f_1(\xi)} \frac{\partial}{\partial \xi} \left(\frac{1}{f_1(\xi)} \frac{\partial}{\partial \xi} \right) + \frac{1}{f_2(\eta)} \frac{\partial}{\partial \eta} \left(\frac{1}{f_2(\eta)} \frac{\partial}{\partial \eta} \right) \right] \Phi(\xi, \eta) = \frac{h_\xi h_\eta}{f_1(\xi) f_2(\eta)} s(\xi, \eta). \quad (111)$$

This equation is now directly solvable via cyclic reduction. This remarkable fact stems from no approximation, but from the “nice” properties of the confining magnetic field. Addition of a third coordinate via rotation about an axis of symmetry introduces terms proportional to the inverse aspect ratio, which may be solved iteratively, as in the text.

B Charge conservation by the filter

Here we consider whether the general metric filter operation conserves charge; i.e. if $\int (\mathcal{S} \circ \rho) dV = \int \rho dV = Q$ where \mathcal{S} is the filter operator and Q is the total charge. Although it is usually taken for granted that the filter operator conserves charge, in a non-Cartesian system this may not be generally true.

We examine the effect of a simple digital filter on the total charge. For simplicity, we choose a single application of the binomial digital filter for our flat-metric filter operator, and consider only a single dimension. From Eq. 70, we have

$$\begin{aligned} (\mathcal{S} \circ \rho)_i &= \frac{1}{J(\xi_i)} \frac{\sigma_{i-1} + 2\sigma_i + \sigma_{i+1}}{4} \\ &= \rho_i + \frac{\Delta^2}{4J(\xi_i)} \sigma_i'' + \mathcal{O}(\Delta^4) \end{aligned}$$

using the three point approximation to the second derivative. If we neglect the $\mathcal{O}(\Delta^4)$ contribution and assume the continuum limit, the conservation of charge equation becomes

$$\begin{aligned} \int (\mathcal{S} \circ \rho) J(\xi) d\xi &= \int \rho J(\xi) d\xi + \frac{\Delta^2}{4} \int \frac{\partial^2 \sigma}{\partial \xi^2} d\xi \\ &= Q + \frac{\Delta^2}{4} \left. \frac{\partial \sigma}{\partial \xi} \right|_0^{L_\xi} \end{aligned}$$

This additional term on the right hand side is a boundary term and vanishes if even parity or continuity boundary conditions are applied to σ from within the filter (the latter can be shown if the integration over θ is performed).

A remaining concern is whether continuity boundary conditions at the origin are in fact appropriate when interpolating or filtering σ , given that $\rho = \sigma/J$, with J vanishing at $r = 0$. In interpolation, it is necessary that continuity be assumed so that a particle does not “disappear” from the simulation if it passes through the origin. A similar argument can be given for the filter. The apparent conflict can be resolved by noting that interior to the first grid point, the variation of the Jacobian is unimportant to the charge accumulation process—the region bounding the $r = 0$ point can be viewed as simply one large cell.

Of greater consequence is the effect at neighboring grid points, for which density fluctuations cause a large response via the $1/J$ term. This is a concern due to the relative scarcity of particles in the central region; nonphysical fluctuations near $r = 0$ can be minimized (but not completely alleviated) through grid stretching there (cf. Fig. 2). The filter operator usually plays a key role in reducing nonphysical fluctuations, but the filter operator given above (with continuity boundary conditions on σ) is lacking in this regard at $r = 0$. Ideally, one wishes to filter ρ , not σ . Taking $\sigma(r = 0) = 0$ reduces fluctuations, but only at the cost of nonconservation of charge. An intriguing possibility is to enlarge the effective particle size in θ near the origin via the k -space smoothing operator, but this will not affect $m = 0$ fluctuations. Nonetheless, to date we have found the present treatment adequate. Larger than normal fluctuation levels at $r = 0$ can usually be tolerated, as it is often a point of relatively little interest.

References

- [1] D. Schnack and J. Killeen, J. Comp. Phys. **35**, 110 (1980).
- [2] R. E. Denton, J. F. Drake, R. G. Kleva, and D. A. Boyd, Phys. Rev. Lett. **56**, 2477 (1986).
- [3] A. Y. Aydemir, Phys. Rev. Lett. **59**, 649 (1987).
- [4] C. Z. Cheng and H. Okuda, J. Comp. Phys. **25**, 133 (1977).
- [5] J. F. Thompson and Z. U. A. Warsi, J. Comp. Phys. **47**, 1 (1982).
- [6] J. M. Dawson, Reviews of Modern Physics **55**, 403 (1983).
- [7] A. B. Cain, J. H. Ferziger, and W. C. Reynolds, J. Comp. Phys. **56**, 272 (1984).
- [8] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw-Hill, New York, 1985).
- [9] E. K. De Rivas, J. Comp. Phys. **10**, 202 (1972).
- [10] J. D. Hoffman, J. Comp. Phys. **46**, 469 (1982).
- [11] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, New York, 1985).
- [12] T. Tajima and F. W. Perkins, in *Proceedings of the Sherwood Theory Conference*, p. 2p9 (1983).
- [13] T. Tajima, *Computational Plasma Physics: With Applications to Fusion and Astrophysics* (Addison-Wesley, New York, 1989).
- [14] M. Kotschenreuther, Technical Report 360, Institute for Fusion Studies, University of Texas at Austin (1989).
- [15] A. B. Langdon, J. Comp. Phys. **6**, 247 (1970).
- [16] H. Abe, N. Sakairi, R. Itatani, and H. Okuda, J. Comp. Phys. **63**, 247 (1986).

- [17] D. Potter, *Computational Physics* (Wiley, New York, NY, 1973).
- [18] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw-Hill, New York, NY, 1981).
- [19] J. U. Brackbill and H. M. Ruppel, *J. Comp. Phys.* **65**, 314 (1986).
- [20] V. K. Decyk, *J. Comp. Phys.* **56**, 461 (1984).
- [21] A. Y. Aydemir and D. C. Barnes, *J. Comp. Phys.* **53**, 100 (1984).
- [22] M. J. LeBrun, *Drift Wave Simulation in Toroidal Geometry*, PhD thesis, University of Texas at Austin, 1988.
- [23] D. C. Barnes, T. Kamimura, J. Leboeuf, and T. Tajima, *J. Comp. Phys.* **52**, 480 (1983).
- [24] R. D. Sydora, J. N. Leboeuf, D. R. Thayer, P. H. Diamond, and T. Tajima, *Phys. Rev. Lett.* **57**, 3269 (1986).
- [25] J. L. Geary, T. Tajima, J. Leboeuf, E. G. Zaidman, and J. H. Han, *Computer Phys. Comm.* **42**, 313 (1986).
- [26] T. G. Northrop, *The Adiabatic Motion of Charged Particles* (Interscience, New York, 1963).
- [27] G. Arfken, *Mathematical Methods for Physicists, second edition* (Academic Press, New York, 1970).
- [28] R. G. Littlejohn, *Journal of Plasma Physics* **29**, 111 (1983).
- [29] B. D. Fried and S. D. Conte, *The Plasma Dispersion Function* (Academic Press, New York, 1961).
- [30] S. Ichimaru, *Basic Principles of Plasma Physics* (Benjamin/Cummings, London, 1973).
- [31] H. Naitou, S. Tokuda, and T. Kamimura, *J. Comp. Phys.* **38**, 265 (1980).
- [32] F. F. Chen, *Introduction to Plasma Physics* (Plenum Press, New York, 1974).

- [33] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 1986).
- [34] R. D. Hazeltine, Phys. Fluids **17**, 961 (1974).
- [35] D. W. Ross and S. M. Mahajan, Phys. Rev. Lett. **40**, 324 (1978).
- [36] L. Chen and C. Z. Cheng, Phys. Fluids **23**, 2242 (1980).
- [37] M. J. LeBrun and T. Tajima, Technical Report 347, Institute for Fusion Studies, University of Texas at Austin (1988).
- [38] T. J. Schep and M. Venema, Plasma Physics and Controlled Fusion **27**, 653 (1985).
- [39] C. Mercier, Nucl. Fusion **3**, 89 (1963).
- [40] P. A. Bernhardt and J. U. Brackbill, J. Comp. Phys. **53**, 382 (1984).

Figure Captions

1. Toroidal coordinate system (r, χ, ζ) .
2. (a) Example nonuniform grid (r) (b) Plasma profile, and number of particles per cell for cylindrical or toroidal geometry corresponding to grid in (a). Region of interest is in the center.
3. Example schematic of (r, θ) grid (here 64×32).
4. Trajectories of a mirroring particles in a tokamak field, for (a) Lorentz particle, and (b) drift particle. At left is the projection onto the (r, θ) plane; at right is the view from above.
5. Lorentz particle in tokamak field. (a) trajectory in the vertical and horizontal planes. (b) time dependence of dynamical quantities. The approximate upper and lower bounds for each plot (internal units) are $v_{\parallel} : (0.14, 0.20)$, $E_k : (5.9045, 5.9055)$, $\mu_0 : (0.00107, 0.00109)$, $p_{\phi} : (-69.620, -69.05)$, $L_{\phi} : (-0.25, -0.10)$, $\psi : (-60.0, -45.0)$.
6. Drift particle in tokamak field. (a) trajectory in the vertical and horizontal planes. (b) time dependence of dynamical quantities. (same parameters as in Figure 5). The approximate upper and lower bounds for each plot (internal units) are $v_{\parallel} : (0.14, 0.20)$, $E_k : (5.904, 5.908)$, $p_{\phi} : (-69.590, -69.565)$, $L_{\phi} : (-0.22, -0.13)$, $\psi : (-57.0, -48.0)$.
7. Frequency versus perpendicular wavenumber, magnetized plasma wave (thermal run). Simulation values represented by marks, theory by solid curve.
8. Frequency versus perpendicular wavenumber, Bernstein wave (thermal run). Simulation values represented by marks, theory by solid curve. Dashed curve is theoretical frequency without inclusion of finite size particle effect.
9. Particle distribution moments as a function of (r, θ) . (a), (b), (c) $T_{\parallel} - \langle T_{\parallel} \rangle$ at times $t = 0, 500, 4000 \Omega_i^{-1}$, respectively. (d), (e) v_{\parallel} at times $t = 0, 4000 \Omega_i^{-1}$, respectively.

10. Frequency versus radius for the $m = 8$ and $m = 9$ modes. The arrows show the locations of the mode rational surfaces. The banded frequency structure is a manifestation of the toroidal geometry.

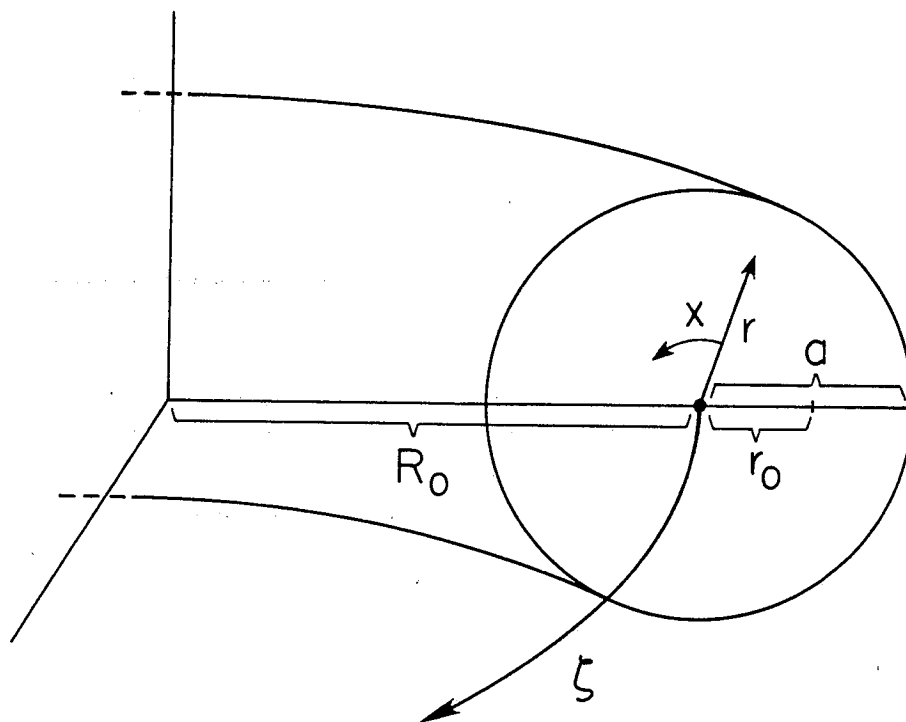


Figure 1

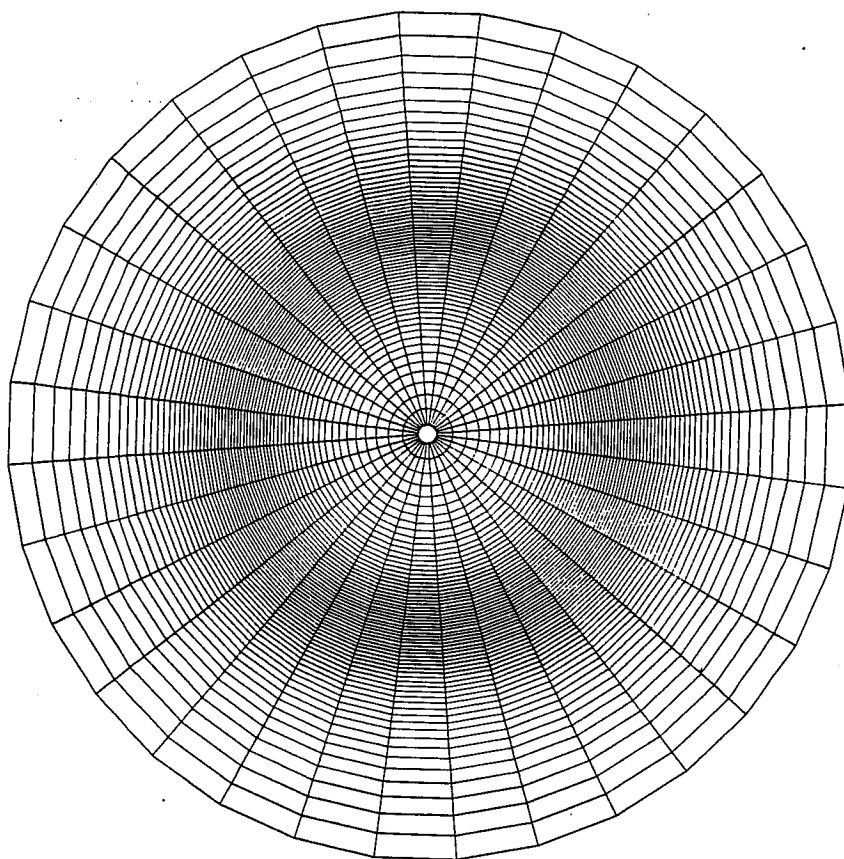


Figure 2

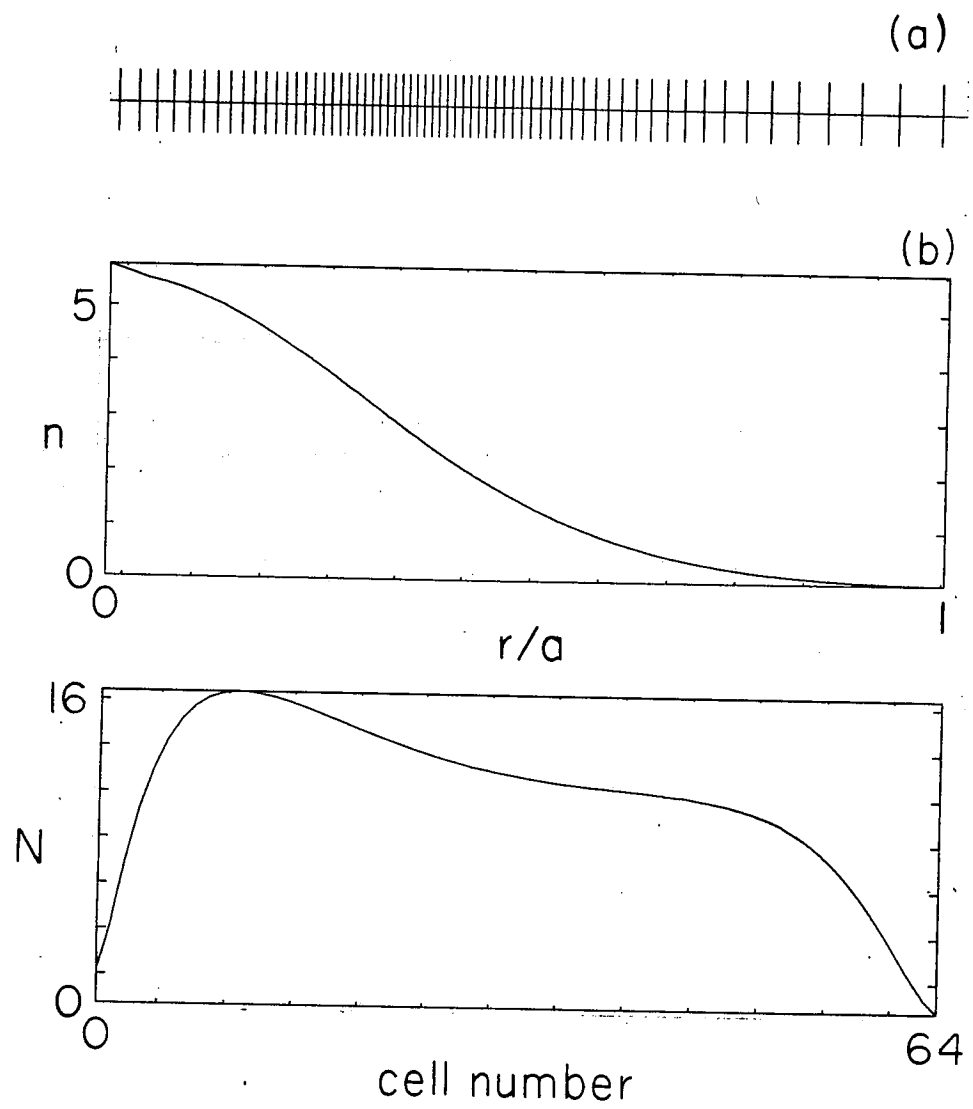
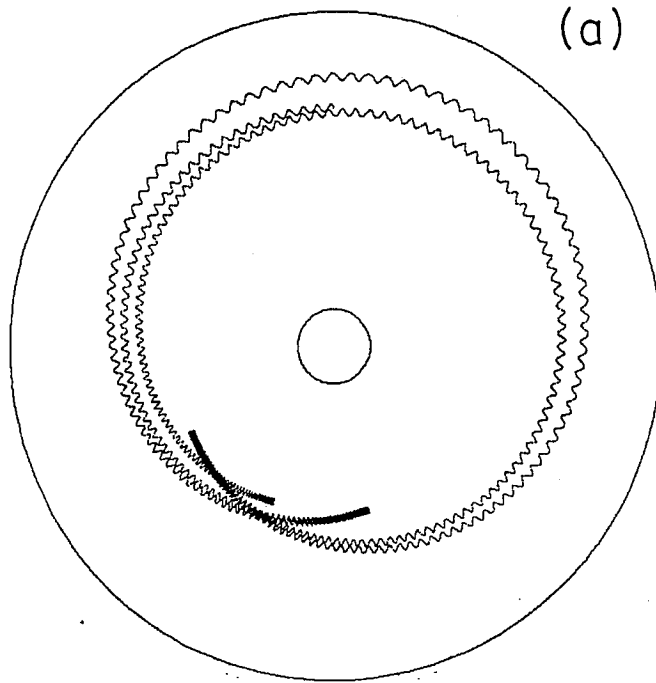
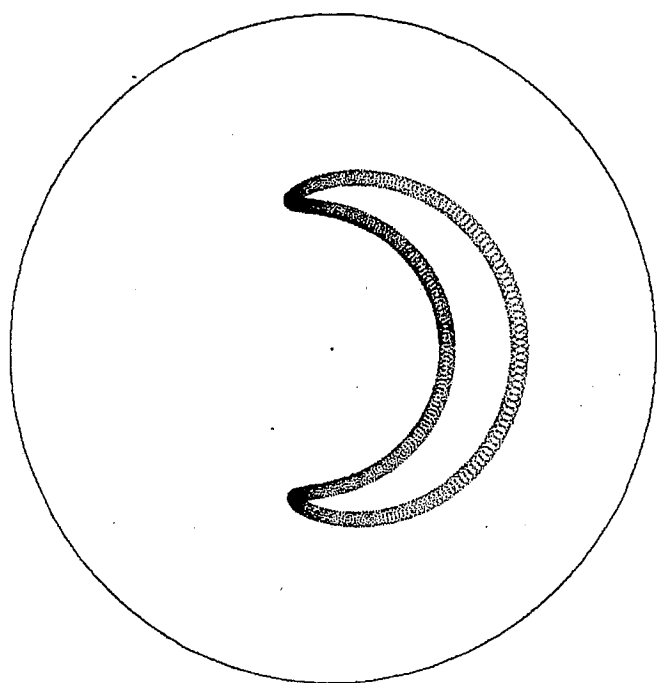
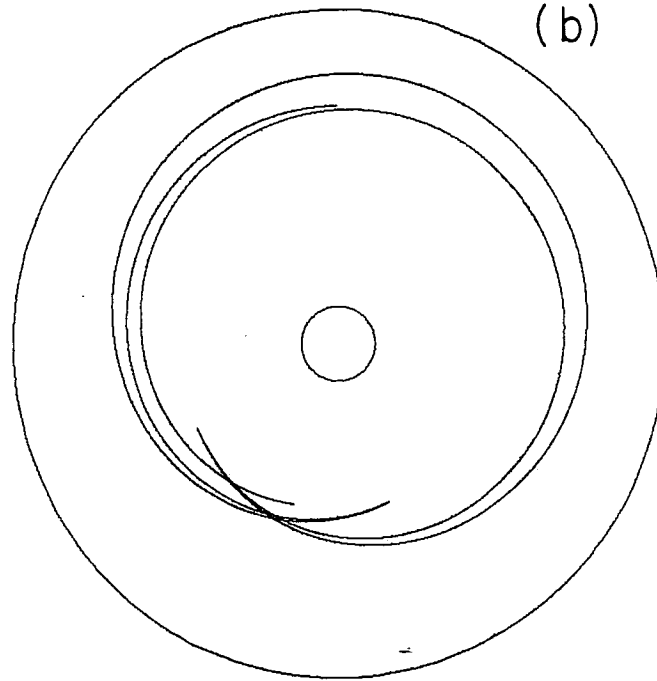
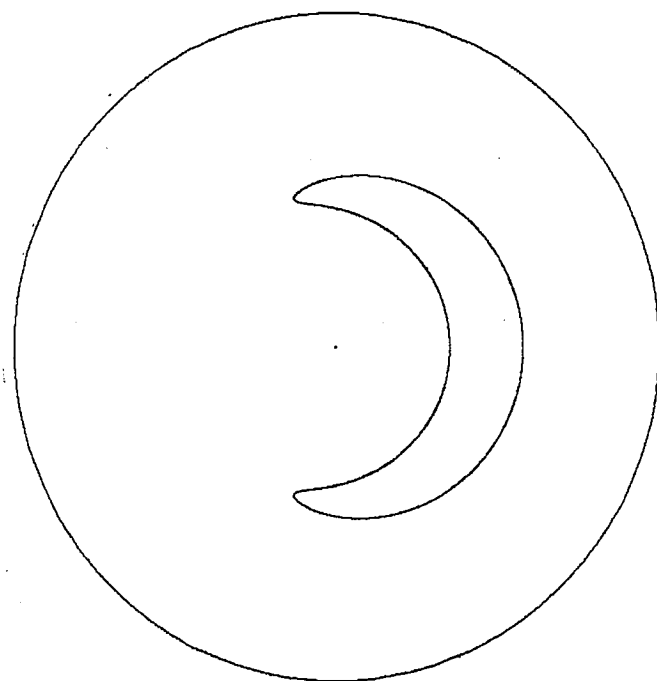


Figure 3



(a)



(b)

Figure 4

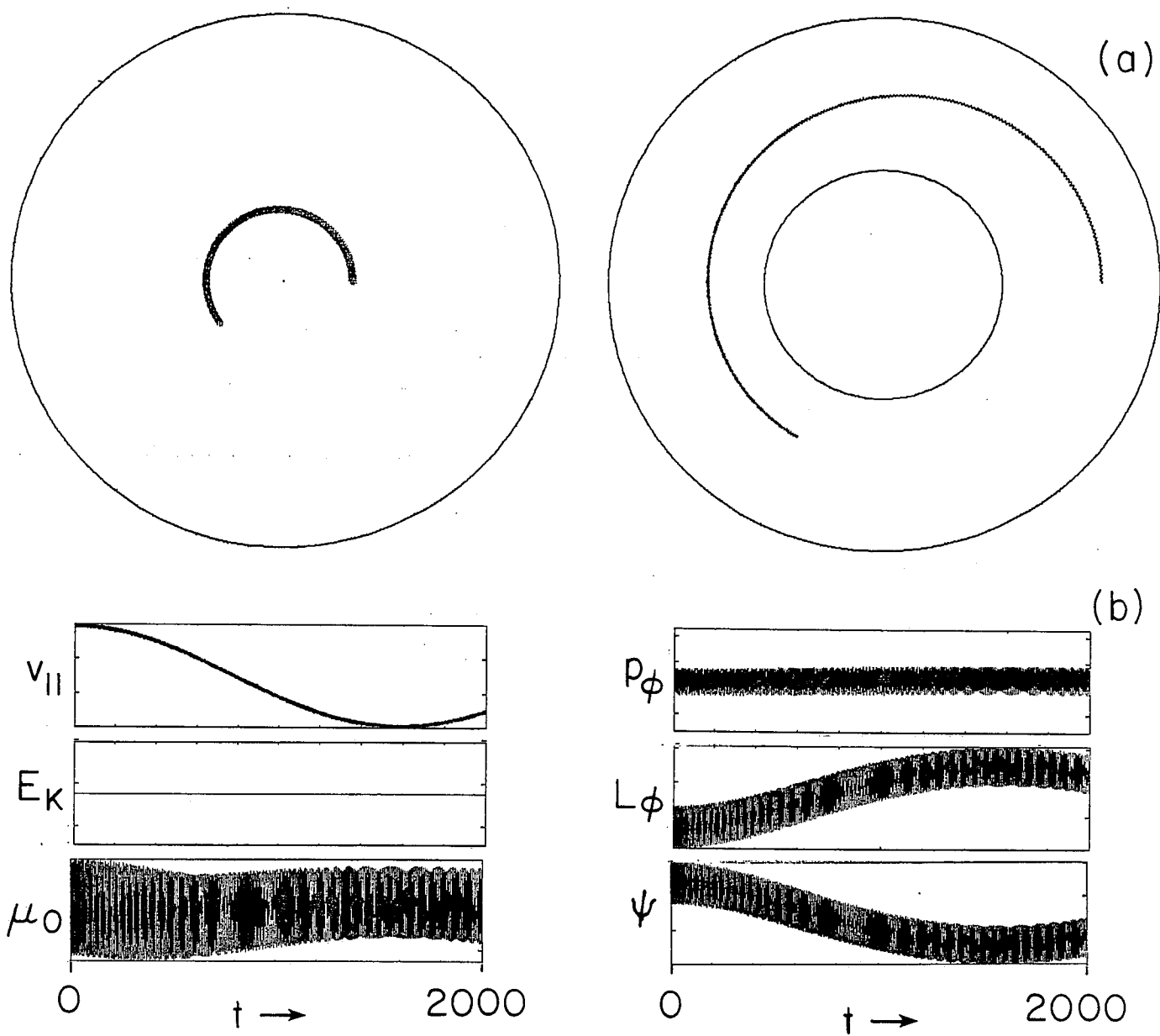


Figure 5

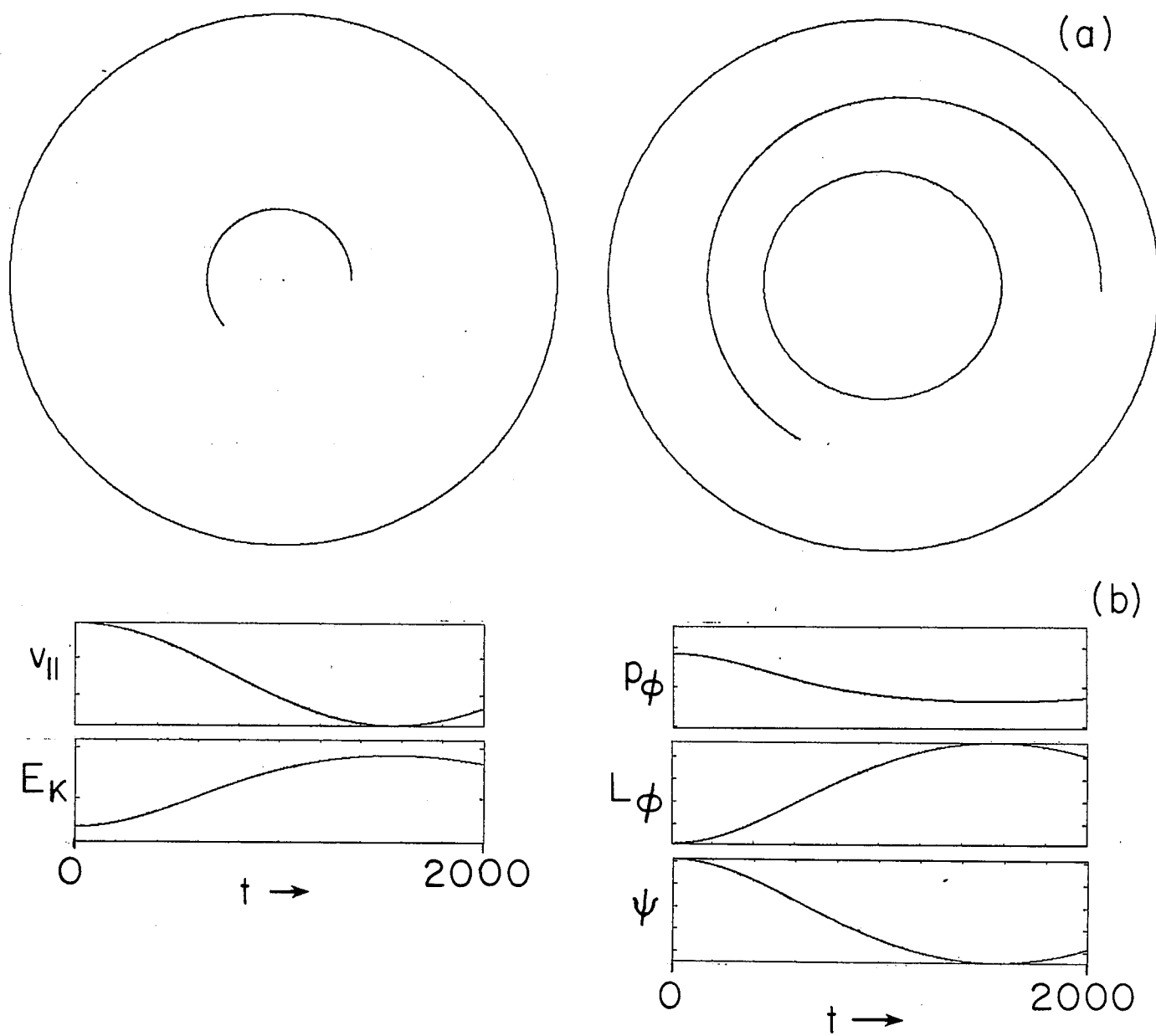


Figure 6

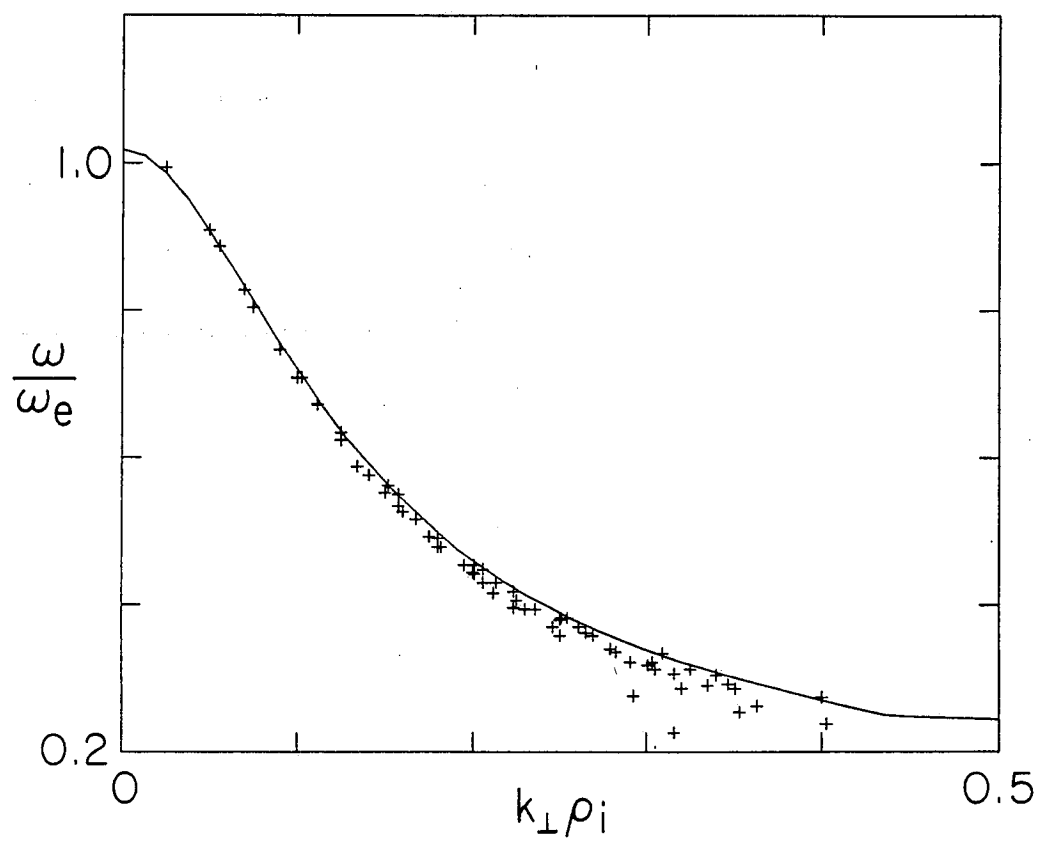


Figure 7

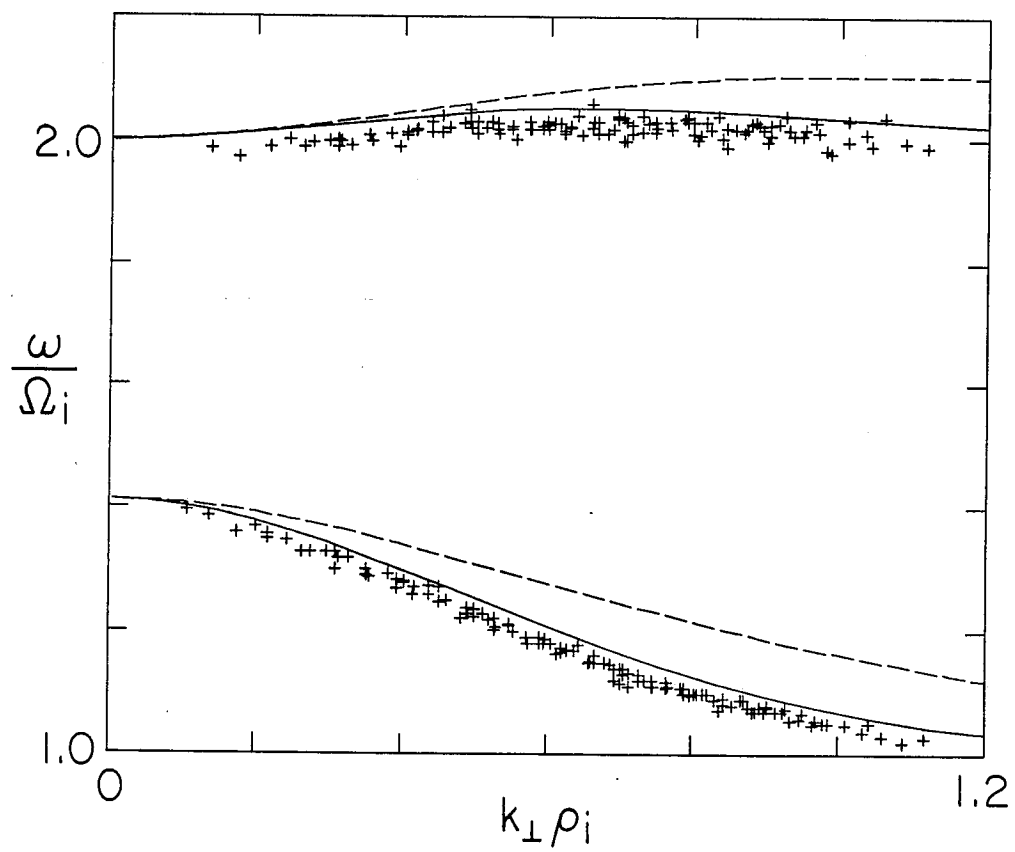


Figure 8

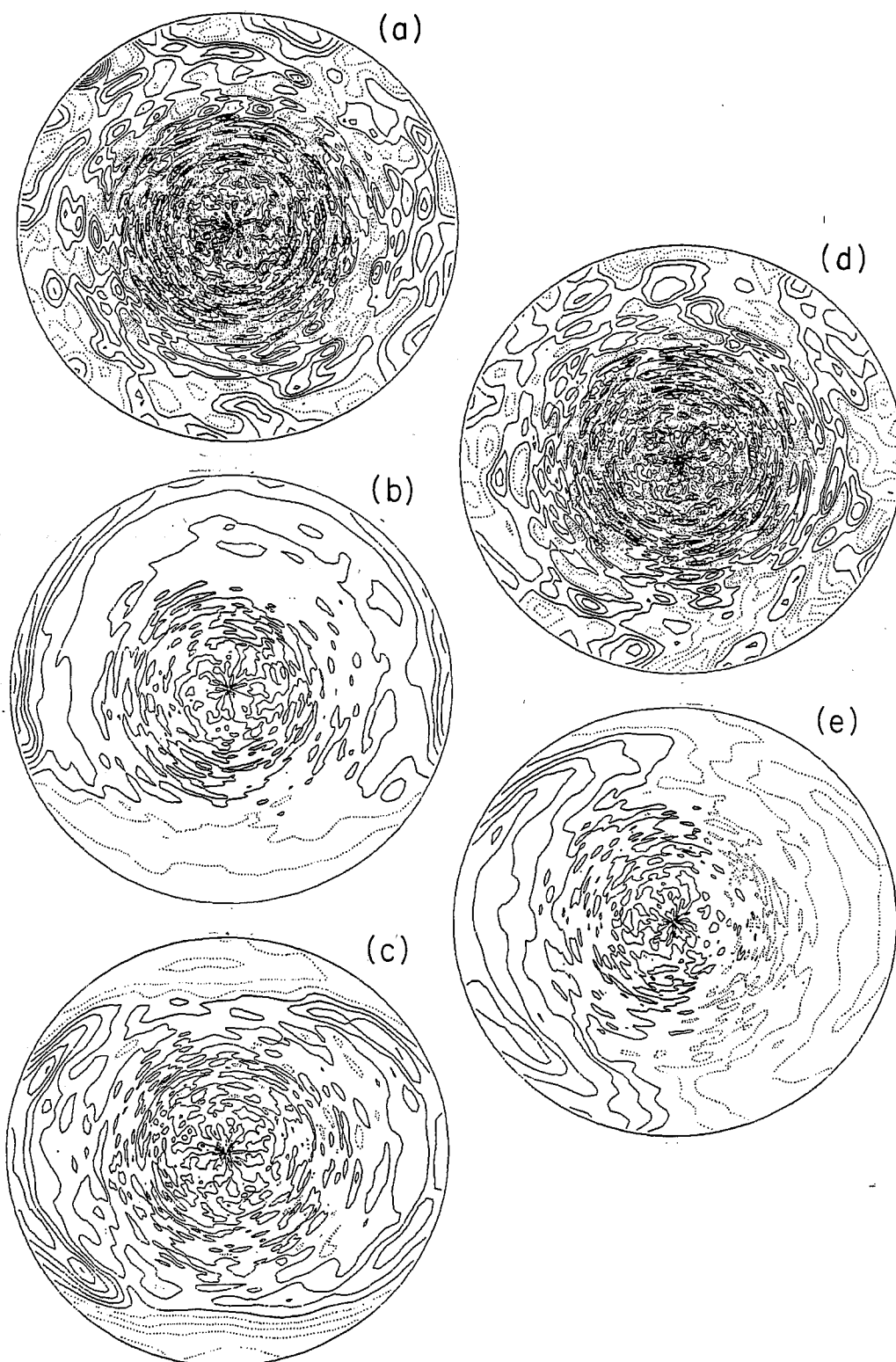


Figure 9

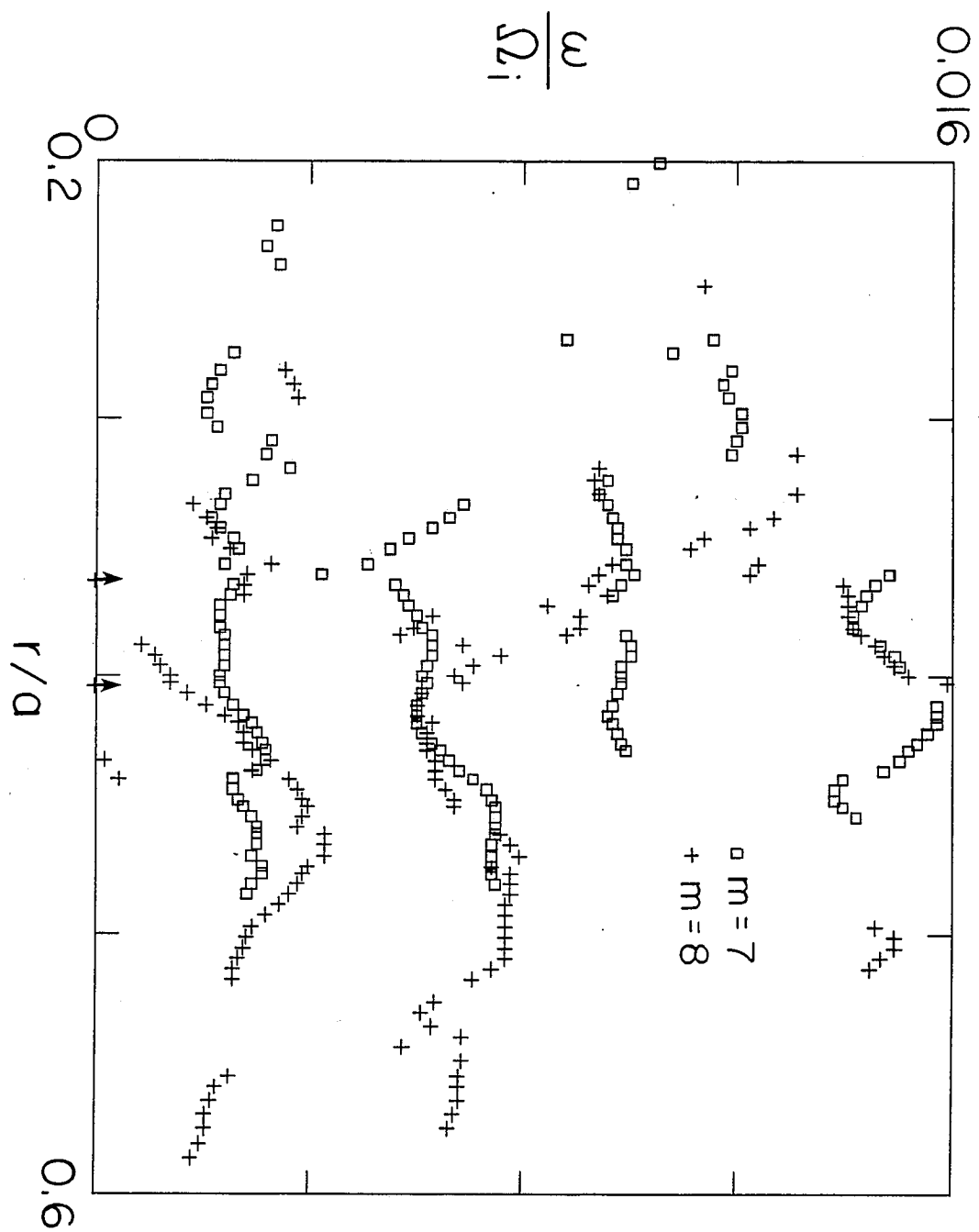


Figure 10