INSTITUTE FOR FUSION STUDIES

DOE/ET-53088-522 IFSR #522

Particle Simulation of Non-Circular Toroidal Plasmas in Non-Orthogonal Curvilinear Coordinates

K. UMEGAKI, M.J. LEBRUN, and T. TAJIMA
Institute for Fusion Studies
The University of Texas at Austin
Austin, Texas 78712

November 1991

THE UNIVERSITY OF TEXAS

AUSTIN

Particle Simulation of Non-Circular Toroidal Plasmas in Non-Orthogonal Curvilinear Coordinates

K. Umegaki, M.J. LeBrun and T. Tajima
Institute for Fusion Studies
The University of Texas at Austin
Austin, TX 78712

Abstract

A numerical algorithm has been developed for electrostatic particle simulation of arbitrarily shaped toroidal plasmas such as axisymmetric tokamaks with non-circular cross-section. Two-dimensional non-orthogonal flux surface coordinates are generated from the numerical solution of the Grad-Shafranov equation. Using a coordinate transformation technique, interpolation and smoothing of charge densities and electric fields are made in the transformed nonorthogonal coordinates. A fast Poisson solver has been developed in which a finite difference scheme for the radial direction and an FFT for toroidal and poloidal directions are employed. A time-decentered Lorentz pusher is used to integrate the momentum equation of ions where decentering is effective to eliminate the ion cyclotron limitation on time step. The Lorentz pushing is carried out in transformed local orthogonal coordinates where two of those coordinates correspond to poloidal and toroidal directions. The ion motion thus contains all electric and magnetic drifts including the polarization drift. Electrons are also pushed in the transformed local orthogonal coordinates using the drift equations. The algorithm is being incorporated into the Toroidal Particle Code.

I. Introduction

Toroidal configurations are important geometry for plasmas and magnetic fields for confinement of hot fusion plasmas. One of the simplest and most commonly studied configurations may be an axisymmetric circular tokamak. This configuration resembles a cylinder in a large aspect ratio (the ratio of the toroidal radius to the poloidal radius). However, in a high plasma beta (the ratio of the plasma pressure to the magnetic one) there arises the Shafranov shift[1] and it deviates from the concentric circular tokamak configuration. There is considerable interest in tokamaks with a noncircular cross-section, as many people believe that such configurations may have improved confinement properties. Included are a doublet configuration, a beam-shape tokamak, and a 'triangular' tokamak.[2] In addition active research is underway for other more complex toroidal geometries[3] such as the stellarators, heliac, reversed field pinch, the geomagnetosphere, the Schwartzschild atmosphere. The kinetic stability and transport of such plasma configurations are vital questions in present thermonuclear fusion research.

We have developed a fully self-consistent particle simulation code for toroidal geometry.[4] The particle simulation has been effective in studying kinetic instabilities and their associated transport properties in a self-consistent fashion.[5,6] In the present Note we present an algorithm for particle simulation of non-circular toroidal plasmas. This is a generalization of our earlier effort.[4] In Ref. 4, the coordinate system was orthogonal curvilinear coordinates for axisymmetric, concentric circular toroidal plasmas. In the present Note we adopt (in general) non-orthogonal curvilinear coordinate systems.

II. Coordinate Systems

A. Magnetic flux coordinate systems

In order to efficiently and/or accurately treat non-circular toroidal plasmas, it is preferable to have a reasonable coordinate system which fit to arbitrarily shaped cross-sections. A possible alternative is to have a brute force 3D cube or 3D cylinder. The trade-off between the general geometry system and the less general ones (cube or cylinder) is the efficacy/accuracy of the code vs. less mathematical sophistication and complexity necessary. We stress the first virtue. This method is similar to the boundary-fitted representation[7] in computational fluid dynamics. Advantages of this include:

(i) Natural control of resolution in the direction of rapidly varying quantities (radial, in this case). Nonuniform mesh may be used to concentrate gridpoints in region of greatest activity;

(ii) Similarly, a good control of grid spacing along magnetic field (typically much larger than across the field). CFL-type conditions on fast-moving particles may be satisfied more easily;

(iii) Representation of simulation fairly close to that of theory, facilitating comparisons;

(iv) Simulation boundary may be chosen to correspond to actual physical boundary, reducing effects of unphysical boundary conditions.

Here, we define 3-D coordinate system which is composed by 2-D non-orthogonal magnetic flux coordinates in the cross-sections and 1-D coordinate in the toroidal direction kept orthogonal to each cross-section. Using this coordinate system, one can realize accurate numerical calculations especially in particle pushing because magnetic field lines (poloidal and toroidal field lines) always coincide with coordinate lines. The 2-D non-orthogonal magnetic flux coordinates are calculated by using a Grad-Shafranov equation[1] solver.

The 3-D non-orthogonal coordinate system (ξ, η, ζ) defined here is shown in Fig. 1. The magnetic flux is constant along the ξ = constant coordinate lines in this system.

B. Non-orthogonal versus orthogonal coordinate systems

It is obvious that orthogonal coordinate system has advantages of simplicity and familiarity. The flux coordinates calculated by Grad-Shafranov solver can be orthogonalized by a variety of techniques. For example, one can apply orthogonal trajectory methods.[8] Figure 2 shows examples of the non-orthogonal and orthogonal grids which we calculated for an elongated cross-section. Orthogonalization may result in a highly distorted grid. These trajectories were calculated from outer to inner boundaries. Even if we change the spacing of the grid at the outer boundary (starting point), it is difficult to avoid grid distortion. Figure 3 shows examples of the grids for elongated and D-shaped cross sections. We can conclude that it is difficult to obtain uniform resolution in orthogonal grid for these geometries. Distortion in grid may substantially reduce timestep in order to satisfy CFL-type condition on fast moving particles.

On the other hand, a non-orthogonal coordinate system has much more flexibility. One can easily control the resolution properties of grid and get reasonably uniform grid spacing along the magnetic field. Although there are some possible concerns as described in the following, we believe that the non-orthogonal coordinate system is superior in principle for non-circular geometry due to its great flexibility. First, extra cross-derivative terms appear in Laplacian of the field equation. These are small if the grid is not severely skewed and are handled iteratively in the field solver along with other iterative terms (those appearing due to the toroidal effects are treated iteratively in TPC code.[4]) Second, finite difference approximation becomes less accurate. However, unless grid is strongly skewed, the accuracy loss is small. Moreover, orthogonal grids are not free from this defect due to the distortion of the grid caused by orthogonalization. Third, it affects FFT calculation in the poloidal

direction η. An FFT in this direction η is straightforward if the grid is uniformly spaced in this coordinate, although the resulting modes are not independent. Coupling between individual modes reduces but does not eliminate the usefulness of a mode representation. If desired, one can use a purely finite difference approach in cross-sectional coordinates (ξ, η).

III. Interpolation and Filtering

A. Interpolation function

Interpolation and filtering are carried out in the transformed natural coordinate system (ξ, η, ζ). The interpolation method adopted in our simulation is that of the area weighting methods. Although we explain the method here in two dimensions for simplicity, applications in three dimensions is straightforward.

We construct a grid of convex quadrilaterals in 2-D non-orthogonal coordinates as shown in Fig. 4. Each quadrilateral cell has 4 vertices at positions $\mathbf{X}_V$. In the coordinate transformation method, we define a natural coordinate system (ξ, η) and map each quadrilateral onto a unit square in the space of natural coordinates. Then, the natural coordinates become integer values at the vertices. The physical (R, Z) coordinates are mapped onto the natural coordinates at the inside of each cell using bilinear interpolation. The interpolation relation can be written as

$$\mathbf{x}_p = [\xi'\{(1-\eta')\mathbf{x}_{i+1j} + \eta'\mathbf{x}_{i+1j+1}\} + (1-\xi')\{(1-\eta')\mathbf{x}_{ij} + \eta'\mathbf{x}_{ij+1}\}] \,, \tag{1}$$

where p is the particle position in the cell and

$$\xi' = \xi - i \,, \quad \eta' = \eta - j \,,$$

$$0 \le \xi', \eta' \le 1 \,.$$

We can define the interpolation function S and write the general equation of interpolation

as

$$\mathbf{x}_p = \sum_v \mathbf{x}_v S(\xi_p - i, \eta_p - j) \ , \tag{2}$$

where v indicates the vertices. This function is used for interpolation of field quantities such as electric $\mathbf{E}$ and magnetic $\mathbf{b}$ fields

$$\mathbf{E}_p = \sum_v \mathbf{E}_v S(\xi_p - i, \eta_p - j) \ , \tag{3}$$

$$\mathbf{B}_p = \sum_v \mathbf{B}_v S(\xi_p - i, \eta_p - j) \ . \tag{4}$$

The same function is used to project particle data onto grid points (vertices). The charge density ρ is calculated by

$$\rho_v = \sum_v \frac{q_p}{J(i,j)} S(i - \xi_p, j - \eta_p) \tag{5}$$

where q_p is the charge and $J(i,j)$ is the transformation Jacobian at the (i,j)th grid point. The interpolation function S satisfies

$$S(\xi', \eta') = (1 - |\xi'|)(1 - |\eta'|) \ , 0 \leq \xi', \eta' \leq 1 \ ,$$

$$S(\xi', \eta') = 0 \ , \text{otherwise} \ , \tag{6}$$

in bilinear interpolation. Since S is a function of $|\xi'|$ and $|\eta'|$, the interpolation operations from particles to grid points become the same between as that from grid points to particles. Here, S is a positive, continuous function with range [0,1] and normalized by

$$\iint_D S \, d\xi d\eta = 1 \ . \tag{7}$$

We can also use quadratic interpolation methods.[9] The method is second-order accurate in the grid spacing, as in bilinear interpolation. However, smoother weighting can be obtained in quadratic methods because both the interpolated function and its first derivative are continuous at the cell boundaries. The additional smoothness results in less error due to

undersampling.[9] This type of interpolation is effective for simulation with large grid spacing (small values of λ_D/Δ). Therefore, we usually adopt quadratic interpolation in the ζ coordinate (along the toroidal magnetic field).

B. Filtering

A commonly-used filter in particle simulations is the convolution of the accumulated charge density with a shape-factor which is performed in k-space of Fourier transformation,

$$\sigma_{sm}(\mathrm{k}) = \sigma_{pr}(\mathrm{k}) \exp(-[\mathrm{k}\cdot\mathrm{a}]^2/2) \ , \tag{8}$$

$$s : \text{smoothed}, \quad p : \text{projected} \ ,$$

where $\sigma = J\rho$, k and a are the wave vector and a measure of the particle size respectively. If we use finite Fourier transform for discretization, we can simply apply this filtering scheme. If we use finite difference scheme for discretization, we can use digital filtering method.[4] An equivalent way of using digital filtering in general non-orthogonal geometry is to add artificial diffusions to variables. For example, if we discretize the equation in 2-D non-orthogonal coordinates using finite difference method, the filtering becomes

$$\sigma_{sm}(\xi,\eta) = \sigma_{pr}(\xi,\eta) + a_{\xi\eta}\nabla^2\sigma_{pr}(\xi,\eta) \ , \tag{9}$$

where $a_{\xi\eta}$: diffusion coefficients. We can apply this digital filtering (smoothing) method for charge density and electric field if desired. The method is equivalent to the treatment of the shape of finite sized particles.

IV. Field Solver

A. 3-D Poisson equation

The field equation for the electrostatic model is the Poisson equation

$$\nabla^2\phi = -4\pi\rho \ . \tag{10}$$

In the cylindrical coordinate system, the Laplacian operator becomes

$$\nabla^2 = \frac{\partial^2}{\partial R^2} + \frac{1}{R}\frac{\partial}{\partial R} + \frac{1}{R^2}\frac{\partial^2}{\partial \theta^2} + \frac{\partial^2}{\partial Z^2} \ . \tag{11}$$

We can derive the Laplacian operator for transformed coordinate system (ξ, η, ζ) defined in Sec. 2. The Poisson equation becomes[10]

$$\frac{1}{J^2}\left(\alpha_{11}\frac{\partial^2\phi}{\partial\xi^2} + 2\alpha_{12}\frac{\partial^2\phi}{\partial\xi\partial\eta} + \alpha_{22}\frac{\partial^2\phi}{\partial\eta^2}\right) + P^*\frac{\partial\phi}{\partial\xi} + Q^*\frac{\partial\phi}{\partial\eta} + \frac{R_0^2}{R^2}\frac{\partial^2\phi}{\partial\zeta^2} = -4\pi\rho \ , \tag{12}$$

where

$$P^* = P + \frac{\beta_{11}}{RJ} \ , \tag{13}$$

$$Q^* = Q + \frac{\beta_{12}}{RJ} \ . \tag{14}$$

The coefficients α_{ij}, β_{ij} are coordinate transformation coefficients and J is the Jacobian. P and Q are coordinate control functions which can be defined using the general 2-D coordinate transformation method[10] as shown in the following.

$$\frac{\partial^2\xi}{\partial R^2} + \frac{\partial^2\xi}{\partial Z^2} = P \ , \tag{15}$$

$$\frac{\partial^2\eta}{\partial R^2} + \frac{\partial^2\eta}{\partial Z^2} = Q \ . \tag{16}$$

Interchanging the dependent and independent variables, Eqs. (15) and (16) becomes

$$\alpha_{11}\frac{\partial^2 R}{\partial\xi^2} + 2\alpha_{12}\frac{\partial^2 R}{\partial\xi\partial\eta} + \alpha_{22}\frac{\partial^2 R}{\partial\eta^2} + J^2P\frac{\partial R}{\partial\xi} + J^2Q\frac{\partial R}{\partial\eta} = 0 \ , \tag{17}$$

$$\alpha_{11}\frac{\partial^2 Z}{\partial\xi^2} + 2\alpha_{12}\frac{\partial^2 Z}{\partial\xi\partial\eta} + \alpha_{22}\frac{\partial^2 Z}{\partial\eta^2} + J^2P\frac{\partial Z}{\partial\xi} + J^2Q\frac{\partial Z}{\partial\eta} = 0 \ , \tag{18}$$

where

$$\alpha_{11} = \vec{\beta}_1 \cdot \vec{\beta}_1 = \beta_{11}^2 + \beta_{21}^2 \ , \tag{19}$$