JOURNAL OF COMPUTATIONAL PHYSICS 52, 209–213 (1983)

Note

Numerical Treatment of the Axial Singularity in a Flux Coordinate System for Particle Simulation

In the realistic simulation of physical systems whether in 2–D or 3–D there often arise situations where the fluid flow has a preferred direction. An example of this is a magnetically confined plasma where the flow is predominantly along the magnetic field lines. To avoid a large numerical diffusion and hence inaccuracy, it is often necessary to adopt a flux coordinate system with coordinates following closely the contours given naturally by the physics of the problem; an example for this is shown in Fig. 1. In this system radial motion is measured by the flux function $\psi = \int \mathbf{B} \cdot d\mathbf{A}$, where $\mathbf{B}$ is the magnetic field and $\mathbf{A}$ an enclosed area whose boundary is traced out by the intersections of a magnetic field line with a poloidal plane. The angular variable θ_0 is measured from the axis which is at the center of the set of nested ψ surfaces. This point at $\psi = 0$ corresponds to multivalues of θ_0 and this causes most general numerical methods to break down. However, we shall show that it is both accurate and efficient in computer time to transform partially to a rectangular system where the singularity clearly does not exist. This differs from standard transformation procedures in that a knowledge of the spatial derivatives of a function, in our case the magnetic field B, in x, y coordinates is not required, hence avoiding a large source of inaccuracy. We show that this is possible by expanding this function B, assuming only its analyticity near the origin, in such a way that possible singular terms are eliminated.

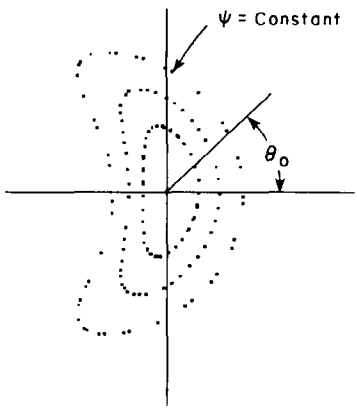

FIG. 1. Schematic diagram of a 2–D flux coordinate system θ_0, ψ.

0021-9991/83 $3.00

We shall use as an example a problem we have encountered in our simulation of a plasma in an asymmetric torus [1, 2]. Here the fact that a flux surface is also an equipotential surface gives additional incentive for the use of flux coordinates. The equations for the drift motion of ions in a magnetic field can be cast in a very simple Hamiltonian form which can be easily integrated [2] if the flux coordinates θ_0, ψ, χ are used. Here ψ is the toroidal flux, θ_0 is a poloidal angle measured from the magnetic axis and χ is a coordinate along the magnetic field line. Since χ does not enter into our essentially 2-dimensional problem we shall henceforth restrict ourselves to θ_0, ψ only.

The drift equations are

$$\begin{aligned} \frac{d\theta_0}{dt} &= -A\frac{\partial B}{\partial \psi} - \frac{\partial \Phi}{\partial \psi}, \\ \frac{d\psi}{dt} &= A\frac{\partial B}{\partial \theta_0}, \end{aligned} \tag{I}$$

where B is the magnetic field and

$$A = \mu + \rho_{\|}^2 B, \qquad \text{where } \mu = \text{the magnetic moment,}$$

$$\rho_{\|} = \text{the parallel Larmor radius,}$$

and

$$\Phi(\psi) = \text{electric potential.}$$

When these equations are integrated very close to $\psi = 0$, the singularity at the axis manifests itself most commonly as the unphysical condition $\psi < 0$ when an integration scheme like the 4th order Runge–Kutta is used. Although the probability of this occurring is in general of the order of the ratio of a few times $\pi \Delta\psi^2$ to the total flux area, where $\Delta\psi$ is the change in ψ in one time step (in our case this ratio is around 10^{-4}), it is very inconvenient especially when particles have long confinement times and the runs abort before data collection is complete.

As this singularity does not exist in Cartesian system we transform to x, y coordinates using circularized flux surfaces:

$$\begin{aligned} x &= (2\psi)^{1/2}\cos\theta_0, \\ y &= (2\psi)^{1/2}\sin\theta_0. \end{aligned} \tag{II}$$

We then advance the x, y coordinates in time instead of the θ_0, ψ coordinates using the equations

$$\begin{aligned} \frac{dx}{dt} &= \frac{1}{(2\psi)^{1/2}}\cos\theta_0\left(\frac{d\psi}{dt}\right) - (2\psi)^{1/2}\sin\theta_0\left(\frac{d\theta_0}{dt}\right), \\ \frac{dy}{dt} &= \frac{1}{(2\psi)^{1/2}}\sin\theta_0\left(\frac{d\psi}{dt}\right) + (2\psi)^{1/2}\cos\theta_0\left(\frac{d\theta_0}{dt}\right). \end{aligned} \tag{III}$$

Substituting Eqs. (I) into (III) gives

$$\begin{aligned}\frac{dx}{dt} &= \frac{A\cos\theta_0}{(2\psi)^{1/2}}\frac{\partial B}{\partial\theta_0} + (2\psi)^{1/2}\sin\theta_0\left[\frac{\partial\Phi}{\partial\psi} + A\frac{\partial B}{\partial\psi}\right],\\ \frac{dy}{dt} &= \frac{A\sin\theta_0}{(2\psi)^{1/2}}\frac{\partial B}{\partial\theta_0} + (2\psi)^{1/2}\cos\theta_0\left[\frac{\partial\Phi}{\partial\psi} + A\frac{\partial B}{\partial\psi}\right].\end{aligned} \tag{IV}$$

At first sight these equations do not appear useful, firstly because of the appearance of $\psi^{1/2}$ in the denominator, and secondly because of the possible singular behavior of $\partial B/\partial\theta_0$, $\partial B/\partial\psi$ which caused breakdown of (I) in the first place. A straightforward transformation to x, y coordinates would require the knowledge of the spatial derivations $\partial B/\partial x$, $\partial B/\partial y$, etc., but in general these are not easily obtainable numerically, often involving interpolation from the original system. The process is therefore both computer time consuming as well as being possibly inaccurate. Assuming only the analyticity of the function B near the origin in x, y system, it may be seen that B must possess a dependence on ψ such that Eqs. (IV) are well behaved near the origin:

$$B = B_0 + \frac{\partial B}{\partial x}x + \frac{\partial B}{\partial y}y + \frac{1}{2}\frac{\partial^2 B}{\partial x^2}x^2 + \frac{\partial B}{\partial x\,\partial y}xy + \frac{1}{2}\frac{\partial^2 B}{\partial y^2}y^2 + \cdots \tag{V}$$

Transforming to polar coordinates with $x = r\cos\theta$, $y = r\sin\theta$ gives

$$B = \sum_m c_m r^l \cos(m\theta), \qquad l \geqslant m,$$

where c_m are constants and we have dropped all antisymmetric terms in our special application. Using analogous arguments for transformation to magnetic coordinates and approximating the residual variations in ψ with a polynomial, we have

$$B = \sum_m \psi^{|m|/2}(a + b\psi + c\psi^2 \cdots)\cos(m\theta_0)$$

from which

$$\frac{\partial B}{\partial\theta_0} = \sum_m m\psi^{|m|/2}(a + b\psi + c\psi^2 + d\psi^3)\sin(m\theta_0), \tag{VI}$$

$$\begin{aligned}\frac{\partial B}{\partial\psi} = \sum_m \Big[&\frac{|m|}{2}\psi^{(|m|/2-1)}(a + b\psi + c\psi^2 + d\psi^3)\cos(m\theta_0)\\ &+ \psi^{|m|/2}(b + 2c\psi + 3d\psi^2)\cos(m\theta_0)\\ &- \psi^{|m|/2}(a + b\psi + c\psi^2 + d\psi^3)\sin(m\theta_0).\end{aligned}$$

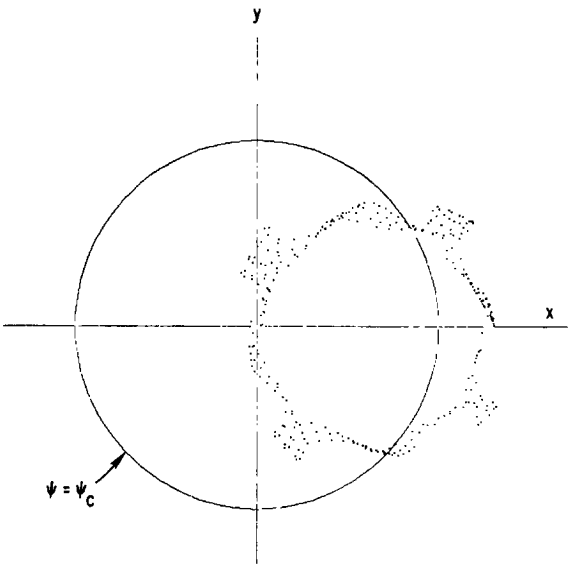

FIG. 2. Projections of a trajectory of a particle in an asymmetric toroidal plasma device. Points inside the circle $\psi = \psi_c$ are calculated in (x, y) system and points outside in (θ_0, ψ) system.

For $|m| = 1$ the first term of the second equation in (VI) tends to infinity as ψ goes to 0. This is of course the reason why Eqs. (I) are not useful near the origin. However, when (VI) is incorporated into the Cartesian formulation in (IV) the multiplying factor $\psi^{1/2}$ removes the singularity in the first term of $\partial B/\partial\psi$.

We have implemented the algorithm outlined above in (ψ, θ_0) coordinates. Figure 2 shows the projected trajectory in θ_0, ψ plane of a particle whose path straddles the boundary at $\psi = \psi_c$. For $\psi < \psi_c$ the particle position is advanced in x, y space while for the region $\psi_c < \psi < \psi_a$ the computation is entirely in θ_0, ψ space. Here $\psi_c = 10^{-4}$ and $\psi_a = 1$. The choice of ψ_c depends mostly on the maximum $\Delta\psi$ the particle makes per time step in the region around $\psi = 0$. To prevent the particle from crossing the axial region in one step starting from $\psi > \psi_c$, it is desirable to make $\psi_c/\Delta\psi \gtrsim 10$.

Major alteration to the computer program is only required in the subroutine supplying the RHS of (IV) to the Runge–Kutta control subroutine. In order to retain as much as possible the vectorization benefits of the CRAY-I compiler, we make the same array store either (θ_0, ψ) or (x, y) depending on the value 0 or 1 of a switch. This switch is used to direct the flow to either the (θ_0, ψ) or the (x, y) blocks of code. Testing on ψ to find out if the particle has just entered or left the x, y computational region given by $\psi < \psi_c$ is performed once per time step in the main program.

An alternative solution when Eqs. (IV) are only used when $\psi_c \ll 1$ is to drop all higher terms in the polynomial for B, (V), keeping only the constant term. This was found to reduce the computer time required for one time step by 20%.

ACKNOWLEDGMENTS

This work was supported by United States Department of Energy Contract DE–AC02–76–CHO–3073. I wish to thank A. H. Boozer and C. F. F. Karney for their helpful discussions.

REFERENCES

1. G. KUO–PETRAVIC, A. H. BOOZER, J. ROME, AND R. FOWLER, *J. Comput. Phys.*, in press.
2. A. H. BOOZER AND G. KUO–PETRAVIC, *Phys. Fluids* **24** (1981), 831.
3. A. H. BOOZER, *Phys. Fluids* (1982).

RECEIVED: October 13, 1982; REVISED February 2, 1983

GIOIETTA KUO–PETRAVIC

Plasma Physics Laboratory
Princeton University P.O. Box 451
Princeton, New Jersey 08544