JOURNAL OF COMPUTATIONAL PHYSICS 87, 349–365 (1990)

Computation of Zero β Three-Dimensional Equilibria with Magnetic Islands

A. H. REIMAN AND H. S. GREENSIDE*

Plasma Physics Laboratory, Princeton University, Princeton, New Jersey 08543

Received November 21, 1988; revised March 22, 1989

A Picard iteration scheme has been implemented for the computation of toroidal, fully three-dimensional, zero β equilibria with islands and stochastic regions. Representation of the variables in appropriate coordinate systems has been found to be a key to making the scheme work well. In particular, different coordinate systems are used for solving magnetic differential equations and Ampere's law. The current profile is adjusted when islands and stochastic regions appear. An underrelaxation of the current profile modifications is generally needed for stable iteration of the algorithm. Some examples of equilibrium calculations are presented.

I. INTRODUCTION

Magnetic island formation is a fundamental concern in toroidal magnetic confinement devices for high temperature plasmas. In otherwise axisymmetric devices, such as tokamaks, tearing modes and field errors lead to island formation. The presence of large islands can result in a deterioration of the confinement properties of the magnetic field. Overlap of islands due to tearing modes is believed to be a major cause of disruptions in tokamaks. In three-dimensional devices such as stellarators, even equilibrium effects can produce large islands and destroy flux surfaces.

The primary approach to computing magnetic islands has been to use resistive time-dependent codes which follow the details of the plasma evolution on the inertial time scale. An alternative approach is to solve directly for equilibria with islands. In this paper we describe a fully toroidal calculation of fully three-dimensional, zero β equilibria with islands. Our code does not assume the existence of good flux surfaces. Finite beta island calculations require modifications in our code beyond those needed to handle zero β islands, and such calculations will not be discussed here.

To place our work in context, we briefly review other efforts to develop equilibrium codes that can handle magnetic islands.

Sykes and Wesson [1] have constructed a code which calculates helical equi-

* Permanent address: Department of Computer Science, Duke University, Durham, N. C. 27706.

0021-9991/90 $3.00

libria with islands at zero β (zero plasma pressure) using a large aspect ratio ordering. They have applied their code to solve directly for nonlinearly saturated $m=2$ tearing modes and kink modes in a cylinder. Their code assumes helical symmetry, so that islands can arise only at the rational surfaces having the assumed helical pitch, and flux surfaces exist everywhere.

Betancourt *et al.* [2, 3] have recently reported a helical equilibrium code which can handle islands at finite β. The helical flux function is prescribed, so that the island width is imposed, rather than adjusting self-consistently to an equilibrium without surface currents.

Bateman and Morris [4] have developed a code which calculates toroidal, zero β, nearly axisymmetric equilibria with narrow islands. The narrow island approximation implies that flux surfaces exist everywhere (that is, there are no stochastic regions). They have also applied their code to studies of saturated tearing modes.

Harafuji *et al.* [5] have developed a new three-dimensional MHD equilibrium code which does not assume nested flux surfaces. Their code differs from traditional equilibrium codes in that the pressure and current profiles evolve as the code converges. This approach does not allow the pressure and current profiles to be specified and, therefore, precludes many of the applications that are of interest for a three-dimensional equilibrium code that handles islands. For example, it is of interest to compare saturated island widths as the profiles are varied in a systematic fashion, or to vary a parameter such as the aspect ratio with the profiles held fixed. Alternatively, to follow evolution on a long time scale, one would like to control the current and pressure profiles in a manner dictated by a transport code.

A similar observation may be made concerning time-dependent resistive MHD codes. Even if a fully toroidal time-dependent code were to be developed which is numerically stable for time steps large compared to the Alfvén time scale, the evolution of the current and pressure profiles due to the numerical viscosity and resistivity in such a code would still be an issue.

A previous paper [6] has given an overview of our PIES three-dimensional equilibrium code. No self-consistent island calculations were discussed. Subsequent papers have given detailed discussions of our method for solving magnetic differential equations [7], and of the convergence properties of the code for equilibria having nested flux surfaces [8]. The code has been applied to the calculation of rippled tokamak equilibria with nested flux surfaces [9].

The self-consistent treatment of islands has required development of some new numerical machinery and has also required some modifications of the basic algorithm [10]. In this paper we describe the modified PIES code, discuss why the modifications were needed, and show how the code works for some particular examples. Finite β islands require more extensive modifications than those needed to handle zero β. Finite β islands are the subject of ongoing research and will not be discussed in this paper, aside from some discussion of what further modifications are necessary to handle them.

To make this paper self-contained, we give an overview of the algorithm of the

PIES code in Section II. In that context we introduce the modifications to the algorithm that have been found necessary to handle islands. This serves as an introduction to the following two sections, where the required modifications are described in detail and where particular numerical solutions are discussed.

Section III describes the modified code. Different coordinate systems are now used to solve the magnetic differential equations and Ampere's law. The current profile is adjusted when islands and stochastic regions appear. An underrelaxation of the current profile modifications has been found necessary.

In Section IV we present some examples of equilibrium calculations. An analytically soluble cylindrical equilibrium with a large island is used to benchmark the code. The convergence of the code is illustrated for fully toroidal equilibria with islands, including a fully three-dimensional, fully toroidal, nonlinearly saturated tearing mode.

II. Overview of the Algorithm

Our iterative algorithm for computing three-dimensional equilibria begins with an initial guess for the magnetic field. Typically we use an analytic solution of the equilibrium equations in cylindrical geometry for this purpose. Field lines are followed, and the Cartesian coordinates of the points along the field lines are used to compute magnetic coordinates [7]. The pressure driven currents are calculated using magnetic coordinates. The magnetic field is updated by solving Ampere's law [6, 8],

$$\nabla \times \mathbf{B} = \mathbf{j}, \tag{1}$$

in the presence of those currents. We cycle through this loop until the magnetic field converges.

Our algorithm can be summarized by the equation

$$\nabla \times \mathbf{B}^{n+1} = \mathbf{j}(\mathbf{B}^n). \tag{2}$$

The magnetic field at the $(n+1)$th step is calculated by solving Ampere's law, using a current which is a complicated nonlinear function of the magnetic field at the nth step. In this form, it becomes clear that our method is closely related to the well-known Picard iteration scheme for solving the Grad–Shafranov equation in two dimensions. This iterative scheme for solving the three-dimensional MHD equilibrium equations was proposed by Grad [11] and by Spitzer [12] in the 1950s.

To completely determine our algorithm, we need to say how we are going to specify the current profile in the islands and the stochastic regions. For the zero β case we consider in this paper, it follows from the equilibrium equation,

$$\mathbf{j} \times \mathbf{B} = 0, \tag{3}$$

that **j** must be everywhere parallel to **B**,

$$\mathbf{j} = \lambda \mathbf{B}. \tag{4}$$

It follows from the divergence-free nature of **j** and **B** that λ must be constant along the field lines,

$$\mathbf{B} \cdot \nabla \lambda = 0. \tag{5}$$

On the good flux surfaces, λ is a function of the flux surface. In the stochastic regions, λ must be a constant. The current profile must therefore be adjusted as we iterate the code. When a stochastic region appears, we flatten λ in that region. The λ profile on the good flux surfaces is adjusted so that λ remains a continuous function of position. That is, the value of λ on the flux surfaces forming the boundary of a stochastic region must be the same as that in the stochastic region.

In the islands, λ can in general be an arbitrary function of the flux surfaces. For a saturated tearing mode, $\eta\lambda$ must be flat across the island, where η is the resistivity [13]. If η is flat there, then so is λ. For the work reported here, we have taken λ to be constant in the islands. This is a physically reasonable current profile in the islands, which was also assumed by Sykes and Wesson [1]. (We emphasize that, although we can use the code to solve for saturated tearing modes, the convergence algorithm for the code is not intended to bear any relation to the time evolution of the tearing mode.)

The code uses an analytical specification of λ as a function of the good flux surfaces. As the code iterates, it adjusts the current profile to make λ flat in the islands and the stochastic regions.

To adjust the λ profile, we need to distinguish between the good flux surfaces enclosing the magnetic axis, and the islands and stochastic regions. We have a diagnostic that does that for us [6]. (For the work described in this paper, we do not need to distinguish between the islands and the stochastic regions, since we flatten λ in both regions.) The diagnostic makes use of our algorithm for constructing magnetic coordinates. That algorithm works only on the good flux surfaces enclosing the magnetic axis. Given the information obtained in following a field line, the algorithm gives us an expression for the corresponding flux surface. We apply the algorithm to each field line that we follow. We then calculate the mean square deviation of the field line from the reconstructed flux surface. On the flux surfaces enclosing the magnetic axis, we find that the deviation is quite small, typically of order 10^{-5}. (The magnitude of this quantity is controlled in the code by the number of Fourier modes retained and by the parameter [7] *ftprec* that specifies the accuracy of the Fourier decomposition along field lines.) In the islands and stochastic regions we find that the deviation is of order 1. As described in Ref. [6], we have tested this diagnostic on fields having large islands and ergodic regions, and have found that it sensitively distinguishes the flux surfaces enclosing the magnetic axis from the islands and stochastic regions.

Our initial island computations using the PIES code encountered numerical difficulties. The source of these problems was determined to be the nature of the

magnetic coordinate system in the neighborhood of a separatrix. For three-dimensional equilibria with nested flux surfaces [8, 9], the code used a magnetic coordinate system. When islands appeared, it used a "quasimagnetic" coordinate system, with the coordinates in the islands calculated by interpolating between the flux surfaces outside the island. These coordinates have rapidly increasing Fourier components near a separatrix, which lead to problems in the numerics. To solve this problem, the code was modified to transform to a different coordinate system before entering the Ampere's law solver. The problem and its solution will be described in more detail in Section III.

An additional problem was introduced by the current profile adjustments in the presence of islands and stochastic regions. This sometimes led to difficulties with convergence, with the island width oscillating from one iteration to the next. The problem was cured by introducing an underrelaxation in the modification of the current profile. Section III also describes this problem and its solution in more detail.

III. The Modified Code

In this section we describe the modifications introduced in the code to handle islands and stochastic regions. A coordinate transformation has been implemented before the Ampere's law solver. The code now adjusts the current profile when islands and stochastic regions appear. An underrelaxation of the current profile modifications has been introduced. We first describe the coordinate modifications.

In the initial formulation of the algorithm for the PIES code [6] it was recognized that it would be advantageous to use different coordinate systems for solving the magnetic differential equations and Ampere's law. Magnetic coordinates are desirable for solving the magnetic differential equations. In these coordinates the magnetic field lines are straight, and Fourier decomposition reduces the magnetic differential equations to algebraic equations. Coordinates of this sort are not optimal for the Ampere's law solver, because they provide an inefficient representation of the magnetic field and the current. The initial implementation of the PIES code nonetheless worked entirely in magnetic coordinates when the flux surfaces were unbroken. When stochastic regions and islands appeared, the coordinates were calculated in those regions by interpolating between flux surfaces. The resulting coordinates behave poorly in the neighborhood of a separatrix. The resonant ripple of the flux surfaces increases rapidly as a separatrix is approached. The lines of constant magnetic angle converge at the x-point. It is not too surprising that numerical calculations in such a coordinate system encounter problems. To cure those problems, we have implemented a transformation to a well-behaved coordinate system before the Ampere's law solver.

The coordinate systems (ρ, θ, ϕ) used in the code are specified by giving the Cartesian coordinates as a function of ρ, θ, and ϕ,

$$\mathbf{x} = \mathbf{x}(\rho, \theta, \phi).$$

All quantities in the code are represented by discretizing in the ρ direction and Fourier decomposing in θ and ϕ. In particular, $\mathbf{x}(\rho, \theta, \phi)$ is specified by a set of Fourier coefficients $\mathbf{x}_{nm}(\rho_j)$ on a set of flux surfaces labeled by $\rho_j = j/L$, where $-N \leqslant n \leqslant N$, $0 \leqslant m \leqslant M$, and $0 \leqslant j \leqslant L$. For all coordinate systems used in the code, ϕ coincides with the geometric toroidal angle. We assume "stellarator symmetry" (symmetry with respect to double reflection in θ and ϕ). This allows us to reduce the number of Fourier modes by a factor of two, and also simplifies things somewhat by guaranteeing that the magnetic axis lies in the midplane at $\phi = 0$.

Each iteration of the algorithm begins by following a set of field lines for the current magnetic field. The code follows $L+1$ field lines, with initial points lying on the intersection of the $\phi = 0$ plane with the midplane, equally spaced between the magnetic axis and the boundary. For magnetic fields with nested flux surfaces, the $L+1$ field lines determine $L+1$ coordinate surfaces. These surfaces are labeled by the coordinate ρ, with equal increments in ρ between the surfaces, $\rho = 0$ at the magnetic axis, and $\rho = 1$ at the boundary. That defines the coordinate ρ. Given the Cartesian coordinates of the points along the field lines, the code applies an algorithm [7] that calculates Fourier coefficients $\mathbf{x}_{nm}(\rho)$ corresponding to a magnetic coordinate system (ρ, Θ, ϕ), with ϕ constrained to be the geometric toroidal angle. For those field lines that are determined to lie in an island or a stochastic region, the value of $\mathbf{x}_{nm}(\rho_j)$ is recalculated by interpolating between the values on the good flux surfaces enclosing the magnetic axis. This defines what we call a "quasimagnetic" coordinate system.

For finite β equilibria, magnetic differential equations must be solved on the good flux surfaces to determine the Pfirsch–Schlüter current. Magnetic coordinates are particularly convenient for this purpose. For the zero β case considered in this paper, we need only impose the homogeneous magnetic differential equation Eq. (5). On the good flux surfaces this is equivalent to imposing the constraint that λ be a function only of ρ. In the islands and stochastic regions we satisfy this equation by taking $\nabla\lambda = 0$. Once λ is determined, the current is given by Eq. (4).

Our Ampere's law solver takes as its input the $\mathbf{x}_{nm}(\rho_j)$ and the contravariant components of $\mathbf{j}$ in the (ρ, Θ, ϕ) coordinate system. It solves Eq. (1) in the given (ρ, Θ, ϕ) coordinate system, and returns the contravariant components of the magnetic field in those coordinates. It makes no assumptions about the nature of the (ρ, Θ, ϕ) coordinate system. In particular, it does not use the fact that (ρ, Θ, ϕ) is a magnetic or quasimagnetic coordinate system.

For one test of the ability of the code to properly handle islands, we use an analytic solution of the equilibrium equations in cylindrical geometry [14]. The magnetic field is given by

$$
\begin{aligned}
B_z &= b_0 J_0(\lambda r) + b_1 I_2(y)\cos(2\theta + kz), \\
B_r &= -b_1(\lambda^2 - k^2)^{-1/2}[kI_2'(y) + 2\lambda I_2(y)/y]\sin(2\theta + kz), \\
B_\theta &= b_0 J_1(\lambda r) - b_1(\lambda^2 - k^2)^{-1/2}[\lambda I_2'(y) + 2kI_2(y)/y]\cos(2\theta + kz),
\end{aligned}
\tag{6}
$$

where

$$y = (\lambda^2 - k^2)^{1/2} r,$$

$k = 1/R$, R is the major radius (a constant to be specified), λ is also a constant which we are free to specify, J_2 is the Bessel function of order 2, I_2 is the modified Bessel function, and the prime denotes differentiation with respect to the argument of the Bessel function. We choose λ so that when $b_1 = 0$ there is a $q = 2$ surface in the middle of the plasma. The magnitude of the resonant $m = 2$ perturbation, and therefore the width of the island at the $q = 2$ surface, is controlled by the value of b_1/b_0. Figure 1 shows a Poincaré plot produced by this field for the parameters $\lambda = 0.967$, $R = 1.$, and $b_1/b_0 = 0.15$.

As one test of the code, we simply start the code with Eq. (6), and we let the code iterate. For the original version of the code, in quasimagnetic coordinates, we verified that the residual $|\mathbf{j} \times \mathbf{B}|$ was initially small, but found that the field moved away from this solution and blew up within a few iterations. The source of the problem was determined to lie in the ρ dependence of the resonant Fourier com-

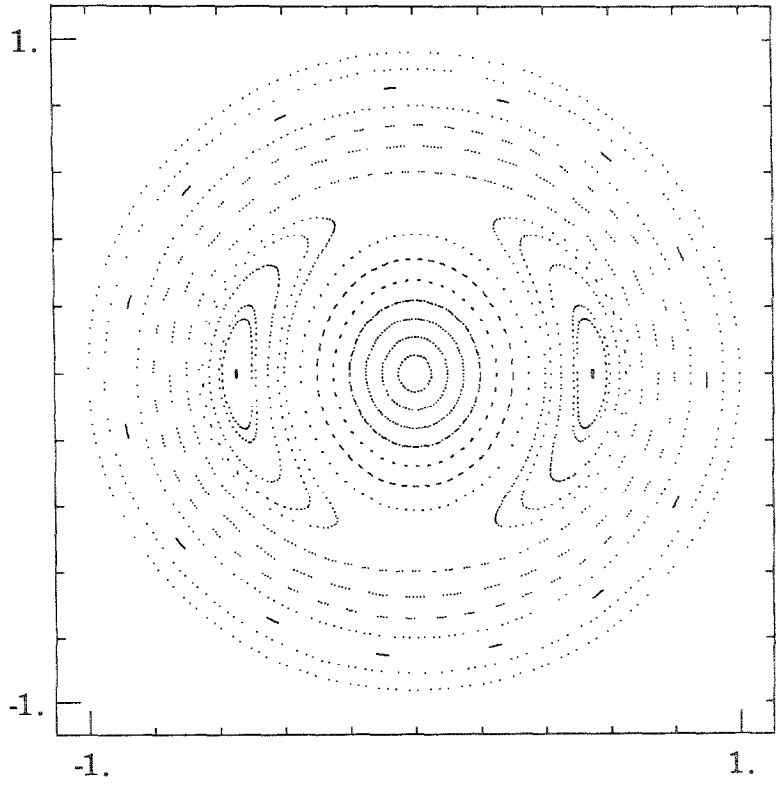

FIG. 1. Poincaré plot for the field given by Eq. (6) with $\lambda = 0.967$, $R = 1.$, and $b_1/b_0 = 0.15$.

ponents of $\mathbf{x}$. These Fourier components increase rapidly in the neighborhood of the island (although they are not singular). The rapid variation leads to large numerical inaccuracies and to numerical instability.

To cure this problem, we have modified the code to transform to a new coordinate system (r, θ, ϕ) before entering the Ampere's law solver. We will call (r, θ, ϕ) the "background" coordinate system, as distinguished from the magnetic or quasimagnetic (ρ, Θ, ϕ) coordinate system used to solve magnetic differential equations. The background coordinates are chosen to be smooth, and they are chosen such that the radial $r=1$ coordinate coincides with the outer flux surface and $r=0$ corresponds to the magnetic axis. The ϕ coordinate is taken to be the same as the geometric toroidal angle (as in the quasimagnetic coordinate system). It is typically the case that the magnetic axis shifts from one iteration to the next, so the condition that $r=0$ coincides with the magnetic axis implies that the background coordinates typically change from one iteration to the next, although the changes become small as the code converges. We will describe in more detail how the background coordinate system is chosen later in this section.

The background coordinate system is specified by giving $\mathbf{x}(r, \theta, \phi)$ in terms of its Fourier coefficients on a set of surfaces $r_j = j/L$, $0 \leqslant j \leqslant L$. Ampere's law is solved in the background coordinate system, giving the updated magnetic field in these coordinates. The field lines are followed in the background coordinate system. Since we know $\mathbf{x}(r, \theta, \phi)$, we can determine the Cartesian coordinates of the points along the field lines. Putting this information into our algorithm for determining quasimagnetic coordinates [7], we obtain an updated $\mathbf{x}(\rho, \Theta, \phi)$ for the new magnetic coordinates.

We need to specify the contravariant components of $\mathbf{j}$ in the background coordinate system for the Ampere's law solver. For the zero β case, we specify the current profile by giving an analytic expression for $\lambda(\rho)$. We analytically prescribe how the λ profile is to be modified in the presence of islands and stochastic regions. The details of this will be discussed later in this section. To determine $\mathbf{j}$ in the background coordinates, we express both λ and $\mathbf{B}$ in these coordinates, and use Eq. (4).

It should now be clear why the zero β case is somewhat easier to handle than the finite β one. To implement the coordinate transformation at finite β we need to map the vector quantity $\mathbf{j}$ onto the new coordinate system. At zero β, we only need to map the scalar quantity, λ.

Given $\lambda(\rho)$, $\mathbf{x}(\rho, \Theta, \phi)$, and $\mathbf{x}(r, \theta, \phi)$, we determine $\lambda(r, \theta, \phi)$ by inverting $\mathbf{x}(\rho, \Theta, \phi)$. For each point in a uniformly spaced (r, θ, ϕ) mesh, we determine $\mathbf{x}$ from $\mathbf{x}(r, \theta, \phi)$, invert $\mathbf{x}(\rho, \Theta, \phi)$ to determine the corresponding value of ρ, and evaluate $\lambda(\rho)$. The inversion need only be performed in two-dimensions, since the ϕ coordinate is known. To perform the inversion, we use a two-dimensional secant algorithm, ZSCNT, from the IMSL package. We have found it advantageous to tinker with the algorithm to take advantage of the known range of the solution $(0 \leqslant \rho \leqslant 1)$ and to allow for immediate convergence if the initial guess is sufficiently accurate. We interpolate $\mathbf{x}(\rho, \Theta, \phi)$ in ρ using cubic splines [7].

The contravariant components of the magnetic field are known in the background coordinate system of the previous iteration. The Cartesian components of the magnetic field are calculated using

$$B^x = \mathbf{B} \cdot \nabla x = B^r \, \partial x/\partial r + B^\theta \, \partial x/\partial \theta + B^\phi \, \partial x/\partial \phi,$$

etc. The Cartesian components can then be expressed as a function of the new (r, θ, ϕ) coordinates much as λ is. For this purpose we need to invert the $\mathbf{x}(r, \theta, \phi)$ function of the previous iteration. We again use a two-dimensional secant algorithm. The covariant components of $\mathbf{B}$ in the new (r, θ, ϕ) coordinates are calculated from

$$B_r = \mathbf{B} \cdot \partial \mathbf{x}/\partial r,$$

etc. Finally, the metric tensor is calculated from $\mathbf{x}(r, \theta, \phi)$ and used to raise the indices of the magnetic field components, giving the contravariant components of $\mathbf{B}$. The contravariant components of the current density are given by Eq. (4).

We choose our background coordinates to coincide with the magnetic axis for $r = 0$, and with the computational boundary for $r = 1$. To accomplish this when the boundary is noncircular, we take

$$\mathbf{x}_{nm}(r) = \mathbf{x}_{nm}(\rho = 1) \, r^m, \tag{7}$$

for $m > 0$, and

$$\mathbf{x}_{n,m=0}(r) = \mathbf{x}_{n,m=0}(\rho = 0) + [\mathbf{x}_{n,m=0}(\rho = 1) - \mathbf{x}_{n,m=0}(\rho = 0)] \, r^2, \tag{8}$$

where the $\mathbf{x}_{nm}(\rho)$ define the magnetic coordinate system of the current iteration, and the $\mathbf{x}_{nm}(r)$ define the background coordinate system. For $r = 0$ and $r = 1$ these background coordinates coincide with the magnetic coordinates. Near the axis, the Fourier coefficients retain the lowest order radial dependence consistent with analyticity of $\mathbf{x}(r, \theta, \phi)$. The radial dependence of the Fourier coefficients is smooth for $0 < r < 1$, with nothing special happening at rational surfaces.

Equations (7) and (8) retain the θ dependence of the magnetic coordinates at the outer boundary, which is less than optimal for representing the magnetic field and the current. We could further modify the coordinates to change the θ dependence, but have not done so because it has not been an issue thus far. For circular boundaries, however, it is trivial to define a background coordinate system with the desired properties and a uniform poloidal angle. For that case, we take the $m > 1$ Fourier components of $\mathbf{x}$ to coincide with those for a cylindrical (r, θ) coordinate system, and again define the $m = 0$ components using Eq. (8).

Putting the magnetic field given by Eq. (6) into the modified code, we again verify that the residual initially calculated by the code is small. Letting the code iterate, we find it now remains in equilibrium. The numerical instability has been cured.

To provide a more rigorous test of the code, we construct a magnetic field which is equal to that given by Eq. (6) at the plasma boundary, but which differs in the interior. Using this for the initial field and imposing the same current profile as before, we verify that the code converges to the analytical equilibrium solution, Eq. (6). The results are described in Section IV.

For the analytical solution discussed thus far, we did not need to flatten λ in the islands because the λ profile is already flat. In general, we do need to flatten λ. To specify $\lambda(\rho)$ in the presence of islands and stochastic regions, it is convenient to define a new radial coordinate $\bar{\rho}(\rho)$ by subtracting from ρ the total enclosed width of the islands and stochastic regions. In other words, $d\bar{\rho}/d\rho = 1$ on the good flux surfaces enclosing the magnetic axis, and $d\bar{\rho}/d\rho = 0$ in the islands and stochastic regions. The coordinate $\bar{\rho}$ is constant in the islands and stochastic regions. If we specify $\lambda(\bar{\rho})$, the λ profile has the desired properties. We are free to adjust the $\lambda(\bar{\rho})$ profile in any way we like as the island widths change. In practice, we use an analytic expression for $\lambda(\bar{\rho})$ with two free parameters which are adjusted to maintain the total plasma current and the current density on the magnetic axis at fixed values.

As an example, one current profile that we have used is the peaked profile of Furth, Rutherford, and Selberg [15],

$$\lambda(\bar{\rho}) = \lambda_0(1 + \bar{\rho}^2/\rho_0^2)^{-2}. \tag{9}$$

This profile has been used extensively in linear and nonlinear tearing mode studies [13, 15, 16]. Equation (9) contains two free parameters, λ_0 and ρ_0. The value of λ_0 determines the current density at the magnetic axis, which is kept fixed. The value of ρ_0 is adjusted to keep the total current in the plasma constant. For any given ρ_0, the current is calculated by integrating up $dI/d\psi = \lambda$ using a trapezoidal rule. A NAG one-dimensional root finder, c05axf, is used to adjust ρ_0 to the required value.

Having determined $\lambda(\rho)$, we express λ as a function of the background coordinates (r, θ, ϕ), as described above. We calculate the current density using Eq. (4), and we solve Ampere's law in the background coordinate system.

In the next section we will describe some equilibrium calculations using the current profile of Eq. (9). To initialize the code, we superpose a resonant perturbation on an axisymmetric magnetic field. When we first used the algorithm we have described with these equilibria, we found that the magnetic field in the code oscillated wildly from one iteration to the next and did not converge. We have found that an underrelaxation (or blending) of the current profile modifications cures this problem. The code was built with the capability for blending the changes in the field [8]. This underrelaxation of the solution was not itself sufficient to resolve the difficulties encountered with these equilibria.

We blend the current in the background coordinate system. For each value of r on our radial grid and each Fourier component $\lambda_{nm}(r)$, we let

$$\lambda_{nm}(r) = (1 - \alpha)\, \lambda_{nm}^{\text{new}} + \alpha \lambda_{nm}^{\text{old}}, \tag{10}$$

where α is specified as an input parameter to the code. In practice, we find that $\alpha = 0.5$ works well.

IV. Examples and Discussion

In this section we present some examples of equilibrium calculations using the modified code. The analytic equilibrium solution with a large island is used to benchmark the code. Fully toroidal equilibria with islands are also computed, including a fully three-dimensional, fully toroidal, nonlinearly saturated tearing mode.

We first consider the analytic equilibrium solution of Eq. (6). When this solution is put into the code, we verify that the residual $|\mathbf{j} \times \mathbf{B}|$ initially calculated by the code is small. Because of discretization errors, the numerically calculated residual for Eq. (6) is nonzero. Figure 2 shows the log of the residual versus log (L) (where $L+1$ is the number of radial grid surfaces) for $L = 10$, 20, and 30. The straight line drawn in the figure is a least squares fit. If the error is dominated by the radial discretization, we would expect the residual to scale as L^{-2}, since our radial finite differencing is second-order accurate. The least squares fit shown in the figure corresponds to a residual of $0.035L^{-1.95}$. Letting the code iterate, we find that the residual decreases as the code converges to a solution of the discretized equilibrium equation.

To provide a more rigorous test of the code, we construct a magnetic field which is equal to that given by Eq. (6) at the plasma boundary, but which differs in the interior. Using this for the initial field and imposing the same current profile as before, we verify that the code converges to the analytical equilibrium solution, Eq. (6). The desired magnetic field is constructed by taking $b_1 \to b_1(\psi_h/\psi_{he})^p$ in Eq. (6), where

$$\psi_h = \frac{b_0}{\lambda}\left[J_0(\lambda r) + \frac{r}{2R}J_1(\lambda r)\right] - \frac{b_1 r^2}{2y}\left[kI_2'(y) + \frac{2\lambda}{y}I_2(y)\right]\cos(2\theta + kz)$$

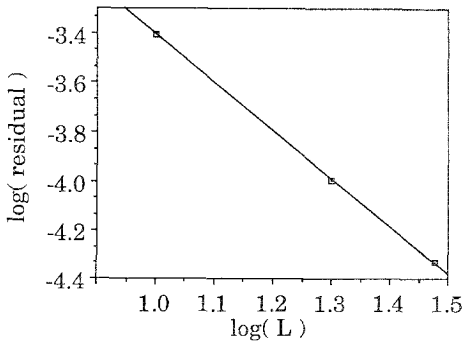

FIG. 2. The log (base 10) of the residual versus log(L), $L = 10$, 20, and 30, for the analytic equilibrium solution. $L+1$ is the number of radial grid surfaces. The straight line is a least squares fit.

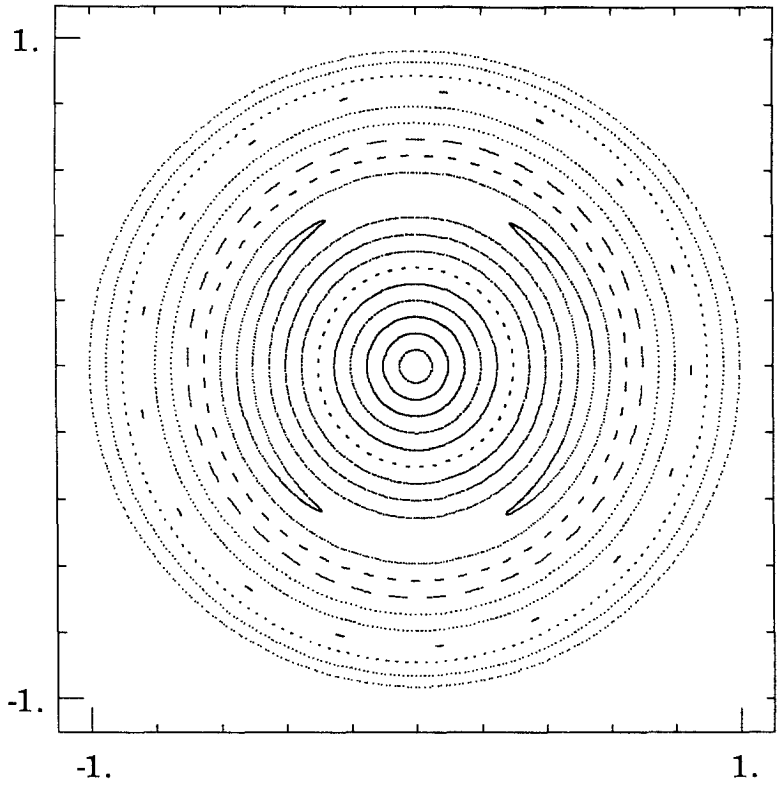

FIG. 3. Poincaré plot for a magnetic field used to initialize the code which agrees with Eq. (6) on the plasma boundary, but differs in the interior.

is the helical flux function corresponding to the field of Eq. (6), ψ_{he} is the value of the helical flux at the plasma boundary, and p is a parameter that controls the initial island width. We choose $p = -500$ to obtain an island width which is initially much smaller than that for the converged solution. The Poincaré plot for the corresponding field is shown in Fig. 3.

Table I gives the residual as a function of the iteration number. For this run, we

TABLE I

Residual as a Function of Iteration Number when the Code Is Initialized with the Field Shown in Fig. 3

Iteration	Residual
0	1.02×10^{-3}
1	1.11×10^{-4}
2	8.55×10^{-5}
3	6.66×10^{-6}

used $L=20$, and we retained the eight largest modes in the range $0 \leqslant m \leqslant 5$, $-2 \leqslant n \leqslant 2$. After the first iteration, the Poincaré plot is already indistinguishable from that of the analytic solution. For this run, the code took about 50 s per iteration. The code has not been fully optimized. We believe that this time can be substantially reduced, and we intend to do so in future versions of the code.

The analytic solution that we have considered has cylindrical geometry and a flat λ profile. We turn now to fully three-dimensional toroidal equilibria. For our current profiles we use the peaked profiles of Furth, Rutherford, and Selberg (Eq. (9)) [15]. These profiles have been used extensively in linear and nonlinear tearing mode studies [13, 15, 16].

Equation (9) has two free parameters, λ_0 and ρ_0. In Ref. [15], Furth *et al.* determine the tearing stability of cylindrical, zero β equilibria as a function of their parameters $x_s \equiv r_s/r_0$ and $x_b \equiv r_b/r_0$, where r is the radial cylindrical coordinate, r_s is the location of the rational surface, and r_b is the location of the conducting wall. For cylindrically symmetric equilibria, our ρ coordinate is the same as r. In our normalization the conducting wall is at $\rho=1$, so that $x_b = 1/\rho_0$. To run any case described in Ref. [15], corresponding to given values of x_s and x_b, we take $\rho_0 = 1/x_b$ and adjust λ_0 to give $\rho_s = x_s \rho_0$.

To initialize the code, we superpose a resonant perturbation on an axisymmetric field. (Recall that the initial field is used only to specify the shape of the boundary, and to serve as an initial guess to the equilibrium field.) The axisymmetric piece of the field is

$$\begin{aligned} B^r &= 0, \\ B^\theta &= (t_0 + t_1 r^2)\, R_0/R, \\ B^\phi &= R_0^2/R^2, \end{aligned} \tag{11}$$

where r and θ are polar coordinates, R_0 is the major radius, and $R = R_0 + r\cos(\theta)$, and where t_0 and t_1 are chosen to give values of t (the rotational transform) at $r=0$ and $r=1$ coinciding with the t in the peaked current analytical equilibrium. The helical perturbation is constructed in terms of an approximate helical flux function,

$$\mathbf{B}_1 = \nabla\psi \times \hat{\phi}. \tag{12}$$

This is only approximately divergence-free because of the toroidal geometry, but that turns out not to matter, and $\nabla \cdot \mathbf{B} = 0$ is enforced by the code on subsequent iterations. We choose ψ somewhat differently for the two cases we consider.

We first consider a current profile having a $q=3$ surface in the interior (where q is the "safety factor," $q = 1/t$). For this case, we take

$$\psi = \varepsilon r^3 \cos(3\theta - \phi) \tag{13}$$

in Eq. (12). We choose $\rho_s/\rho_0 = 0.6$, so that the profile is tearing stable in cylindrical geometry. We run, however, in toroidal geometry, with an aspect ratio of 4. We adjust the parameter ρ_0 so that there is no $q=2$ or $q=4$ surface in the plasma.

Figure 4 shows the convergence as a function of blending parameter for $r_s = 0.8$, $r_0 = 1.33$, and $\varepsilon = 5 \times 10^{-4}$. For these runs, the blending is applied to both the field and the current profile, with the blending parameter the same for both. The figure is a plot of the residual as a function of the iteration number for three different values of the blending parameter, $\alpha = 0.1$, $\alpha = 0.5$, and $\alpha = 0.7$ (where α has been defined in Eq. (10)).

Each of the calculations for Fig. 4 were done with $L = 40$, retaining the 15 largest modes in the range $0 \leqslant m \leqslant 9$, $-1 \leqslant n \leqslant 1$. The Fourier transforms were calculated to within a tolerance of 6×10^{-5} as specified by the input parameter to the code *ftprec*. As in all the calculations described in this paper, we have retained all modes greater than *ftprec*. For each value of the toroidal mode number n, the amplitude of the highest poloidal mode number retained is of order *ftprec*. This amplitude grows larger when the width of the Fourier spectrum increases, prompting us to add more modes. (The same is true of the lowest poloidal mode number when the $m = 0$ mode is not retained.) The coupling in toroidal mode number comes only through nonlinear beating of the modes, so that the amplitudes of the higher n modes are predictably small. We have rerun a few cases with additional modes, and have verified that the change in the mode amplitudes is small. We have also verified the expected amplitudes of the higher n modes.

In calculating the residual, we have used a form of the equilibrium equations different from that used by the convergence algorithm of the code. This provides us with some independent information on our discretization and truncation errors. As in our calculations of equilibria with nested flux surfaces [8], the residual is observed to decrease to a minimum value, and then stop decreasing. When the code has converged, the value of the residual reflects the truncation and discretization errors in the code. The minimum residual was found to scale as $1/L^2$ in Ref. [8], as would be expected if the radial discretization errors dominate.

In the calculations for Fig. 4, the Poincaré plot appears to be converged by the

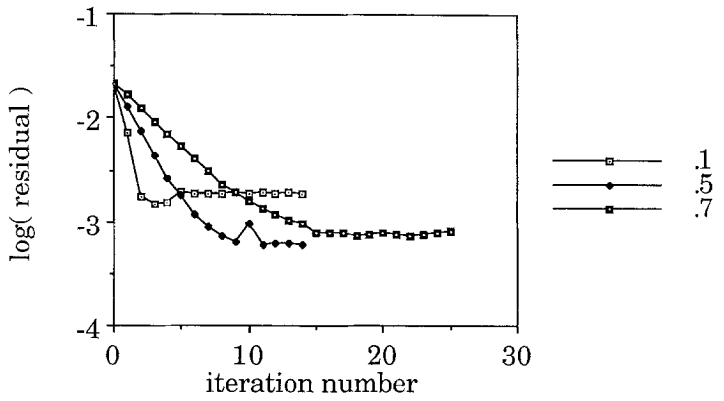

FIG. 4. Plot of the residual as a function of the iteration number for the peaked profile with an $m = 3$ perturbation, $r_s = 0.8$, and $r_0 = 1.33$. Three different values of the blending parameter, α, have been used.

third iteration in each case. At that point the island width has settled down to about 13% of the minor radius, with little change observable after that. The converged island width for the three cases is the same. These runs took about 1.5 min of Cray time per iteration.

We can modify our initial helical perturbation in such a way that it leaves the plasma boundary unperturbed. (The boundary is then circular.) To do that, we take

$$\psi = \varepsilon(r^3 - r^5)\cos(3\theta - \phi). \tag{14}$$

If we put the modified initial field into the code and iterate, we find that the code rapidly converges to an axisymmetric field. This is what we would expect for a tearing stable equilibrium (although the analysis of Furth *et al.* [15], of course, does not guarantee that our toroidal equilibrium is stable). For tearing unstable equilibria, however, there are three-dimensional equilibrium solutions with circular boundary corresponding to the nonlinearly saturated instability.

We next consider a current profile having a $q = 2$ surface in the interior. We take

$$\psi = \varepsilon(r^2 - r^4)\cos(2\theta - \phi), \tag{15}$$

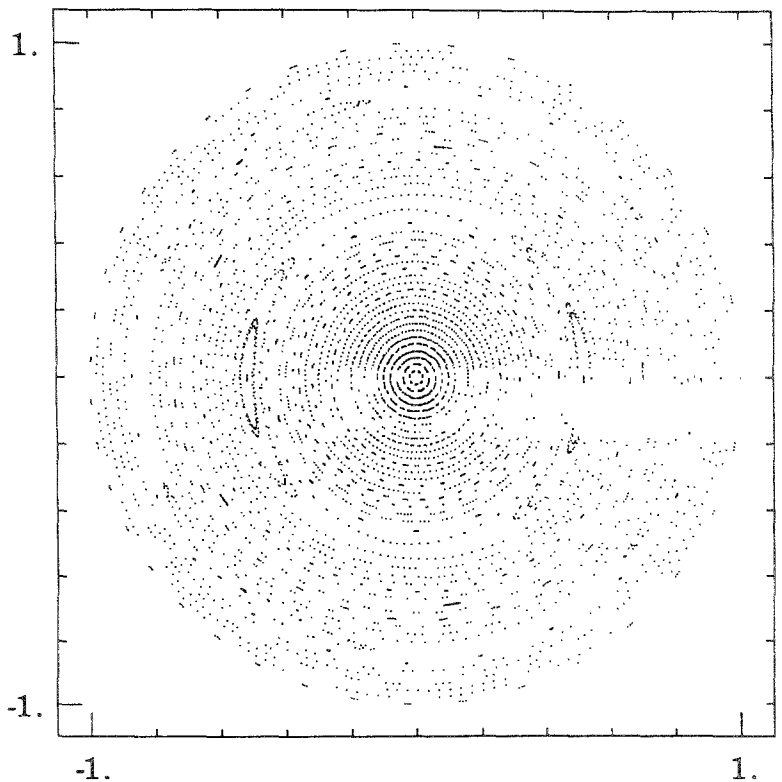

FIG. 5. Poincare plot of the initial field for the tearing unstable case. The boundary is unperturbed.

so that the boundary is unperturbed. For the profile parameters, we take $r_s = 0.525$ and $r_0 = 0.8$, so that the profile is unstable to an $m = 2$ tearing mode in a cylinder and the $q = 2$ surface is about halfway out. Again we take an aspect ratio of 4.

Figure 5 is a Poincaré plot of the initial field, corresponding to $\varepsilon = 1.25 \times 10^{-3}$. Although we have imposed a pure $m = 2$ Fourier harmonic, it couples in toroidal

modes having amplitudes greater than 10^{-5}. Letting the code iterate, we now find that it converges to an equilibrium solution with an island. The Poincaré plot is shown in Fig. 6.

For this calculation we have used cylindrical background coordinates, (r, θ). The origin of the background coordinate system does not coincide with the magnetic axis for the converged solution. This background coordinate system is convenient for producing our Poincaré plots, since the field lines are followed in the background coordinate system.

We have also run a case in which only the amplitude of the initial perturbation is different. The amplitude is taken four times larger, $\varepsilon = 5.0 \times 10^{-3}$, corresponding

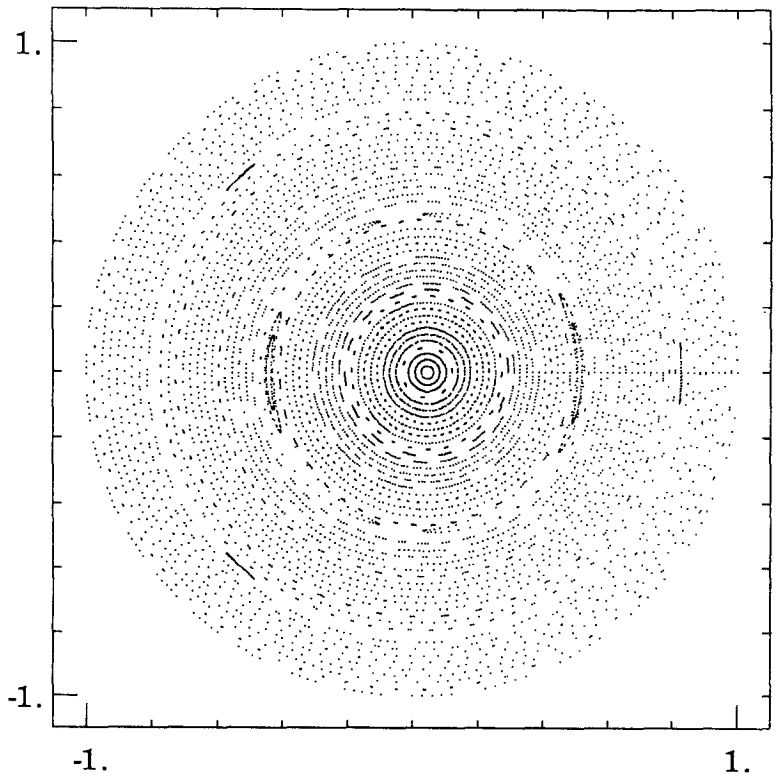

FIG. 6. Poincaré plot for the converged equilibrium field in the tearing unstable case.

to a doubling of the initial island width at the $q = 2$ surface. The code converges to the same final equilibrium solution.

Our previous code often required some radial filtering to stabilize short wavelength modes. This was discussed in Ref. [8]. For the work described in this paper, no filtering was needed. The coordinate transformation has made the code more robust. We have run island calculations out to as many as 45 iterations and have seen no indications of numerical instability.

We conclude that our modified PIES code is capable of calculating fully toroidal, self-consistent, zero β equilibria with islands. The analytic test case indicates that island widths are accurately calculated. The code converges rapidly and robustly.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy under Contract DE-AC02-76-CHO-3073.

REFERENCES

1. A. SYKES AND J. WESSON, in *Proceedings, 8th International Conference on Plasma Physics and Controlled Nuclear Fusion Research, Brussels, 1980* (IAEA, Vienna, 1981), Vol. I, p. 237.
2. F. BAUER, O. BETANCOURT, P. GARABEDIAN, AND M. WAKATANI, *The Beta Equilibrium, Stability, and Transport Codes* (Academic Press, Orlando, FL, 1987).
3. O. BETANCOURT, *Commun. Pure Appl. Math.* **41**, 551 (1988).
4. G. BATEMAN AND R. MORRIS, *Phys. Fluids* **29**, 753 (1986).
5. K. HARAFUJI, T. HAYASHI, AND T. SATO, Hiroshima University Institute for Fusion Theory Report HIFT-132, 1987 (unpublished).
6. A. H. REIMAN AND H. S. GREENSIDE, *Comput. Phys. Commun.* **43**, 157 (1986).
7. A. H. REIMAN AND H. S. GREENSIDE, J. *Comput. Phys.* **75**, 423 (1988).
8. H. GREENSIDE, A. REIMAN, AND A. SALAS, *J. Comput. Phys.*, in press.
9. J. L. JOHNSON AND A. H. REIMAN, *Nucl. Fusion* **28**, 1116 (1988).
10. A. H. REIMAN AND H. S. GREENSIDE, in *Abstracts of the 1987 Annual Controlled Fusion Theory Conference, San Diego, California, April 1987* (S.A.I. Corp., San Diego, 1987), Abstract 2C21; A. H. REIMAN, *Bull. Amer. Phys. Soc.* **32**, 1908 (1987), paper 815.
11. H. GRAD AND H. RUBIN, in *Proceedings, Second United Nations International Conf. on the Peaceful Uses of Atomic Energy* (United Nations, Geneva, 1958), Vol. **31**, p. 190.
12. L. SPITZER, *Phys. Fluids* **1**, 253 (1958).
13. R. B. WHITE, D. A. MONTICELLO, M. N. ROSENBLUTH, AND B. V. WADDELL, *Phys. Fluids* **20**, 800 (1977).
14. S. CHANDRASEKHAR AND P. C. KENDALL, *Astrophys. J.* **126**, 457 (1957).
15. H. P. FURTH, P. H. RUTHERFORD, AND H. SELBERG, *Phys. Fluids* **16**, 1054 (1973).
16. P. H. RUTHERFORD, *Phys. Fluids* **16**, 1903 (1973).