PPPL--2355

DE87 000673

Calculation of Three-Dimensional MHD Equilibria with Islands and Stochastic Regions*

MASTER

A. Reiman and H. Greenside

Plasma Physics Laboratory, Princeton University

Princeton, N.J. 08544

Abstract

A three-dimensional MHD equilibrium code is described that does not assume the existence of good flux surfaces. Given an initial guess for the magnetic field, the code proceeds by calculating the pressure-driven current and then by updating the field using Ampere's law. The numerical algorithm to solve the magnetic differential equation for the pressure-driven current is described, and demonstrated for model fields having islands and stochastic regions. The numerical algorithm which solves Ampere's law in three dimensions is also described. Finally, the convergence of the code is illustrated for a particular stellarator equilibrium with no large islands.

*Presented as an invited talk at the 8th Europhysics Conference on Computational Physics, Eibsee, Fed. Rep. of Germany, May 13-16, 1986. To appear in a special issue of Computer Physics Communications.

1. Introduction

In this paper we describe a three-dimensional MHD equilibrium code that does not assume the existence of good flux surfaces. The code uses an algorithm for solving the equilibrium equations that is different from the energy-minimizing algorithms which have been developed extensively and successfully for 3D equilibrium codes that assume good surfaces[1-3]. We describe the algorithm, our reasons for choosing it, and the numerical methods we have developed to implement it. We also describe the numerical issues raised by the presence of islands and stochastic regions, and how we have dealt with those issues. In particular, we describe how the code solves magnetic differential equations, using as examples model fields with islands and stochastic regions. Finally, we describe the convergence of the code for a particular stellarator equilibrium with no large islands.

We specialize in this paper to the case of zero net current within each flux surface, as is appropriate for applications to stellarators. Our code has been written to deal with the more general case of arbitrary net currents, except for some small modifications that need to be made to deal with the currents in the islands and stochastic regions.

The applications we have in mind for the code are the study of the breaking of flux surfaces in stellarators, and the study of saturated tearing modes in tokamaks and other toroidal confinement devices. The issue that initially motivated our interest in such an equilibrium code is the question of equilibrium beta limits in stellarators.

Stellarator vacuum fields are designed to have relatively good flux surfaces. As pressure is added, currents are driven, consistent with the equation

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

The magnetic field becomes increasingly distorted. It is not known under what conditions the field produced by the pressure-driven currents will cause the formation of large islands and stochastic regions. In particular, it is important to know whether there is an equilibrium beta limit due to such an effect, and what that beta limit is. The convention in evaluating stellarator designs has been to estimate the equilibrium beta limit to be the value of beta at which the magnetic axis shifts halfway to the wall. An analytical calculation of equilibrium island formation in one proposed helical axis stellarator found a much lower beta limit[4].

Nominally axisymmetric devices, such as tokamaks, are in practice generally nonaxisymmetric because of the presence of tearing instabilities, magnetic ripple from discrete coils, and field errors. The problem of finding the 3D equilibrium in the presence of a saturated tearing mode differs from the equilibrium island problem for the stellarator, in that the equilibrium is no longer uniquely specified. The nonaxisymmetric equilibrium bifurcates from a symmetric one at the marginal stability point of the tearing mode. By following the equilibrium as the current profile is modified away from this bifurcation point, it should be possible to calculate saturated island widths. A 3D MHD equilibrium code coupled to a transport code would, in fact, give the full time dependence of island growth in the Rutherford regime. MHD equilibrium solutions with islands are also of interest for studying transport and stability in the presence of saturated tearing modes.

To provide some physical motivation for the computational algorithm we have adopted, we will discuss equilibrium island formation in section 2. In section 3 we describe the algorithm, and discuss our reasons for adopting it. In section 4 we give an overview of our numerical implementation of the algorithm, and we describe some of the mechanics of building the code. The following two sections provide more details on the numerical implementation. Section 5 goes into detail on our numerical solution of magnetic differential equations. Section 6 describes our numerical solution of a three-dimensional Poisson equation in numerically specified coordinates. Section 7 describes the convergence of the code for a finite beta stellarator equilibrium. Finally, in section 8 we present our conclusions.

2. The Physics of Equilibrium Island Formation

In this section we discuss the physics of equilibrium island formation in stellarators. The discussion will provide some physical motivation for the algorithm to be introduced in section 3, and will also serve to clarify what physics our code is intended to describe. A more extended discussion of the physics, focusing particularly on the dominant nonlinear couplings driving the resonant field at the rational surfaces, can be found in ref. [4].

When a small pressure is added to a three-dimensional vacuum field with good surfaces, the magnetic field is perturbed by diamagnetic and Pfirsch–Schlüter currents. The plasma responds initially by developing localized currents at the rational surfaces which shield out the resonant part of the perturbation in **B** there. In the limit of zero resistivity these are delta function currents. Their magnitudes can be determined by a constant Ψ approximation, as in linear tearing mode theory. The resulting field can be written in magnetic coordinates:

$$\mathbf{B} = \nabla\Psi \times \nabla\Theta + \iota \nabla\Phi \times \nabla\Psi. \qquad (2)$$

The localized currents at the rational surfaces decay rapidly in the presence of even small resistivity. As these currents begin to decay, the resonant field is only partially shielded out, and islands begin to open. For small islands, the island width (in Ψ) at the surface where $\iota = m/n$ is

$$w = 4\ \left((\mathcal{J}\mathbf{B}\cdot\nabla\Psi)_{mn} / m\,\iota' \right)^{1/2}, \qquad (3)$$

where $\mathcal{J} = 1/(\mathbf{B}\cdot\nabla\Phi)$ is the Jacobian.

Because of the absence of an externally induced toroidal electric field, stellarators evolve to an equilibrium having zero net current within each flux surface. In particular, the net current inside each island decays to zero. In the absence of heat deposition in an island, or energy loss due to impurity radiation there, the pressure evolves to a flat profile in the island. In that case the local current density is zero everywhere in the island. The island evolves to an equilibrium width determined purely by external currents. More generally, there may be heating in the islands due to neutral beams, or if the island is near the edge there may be a large concentration of impurities there. There are then pressure-driven currents within the island itself which modify the island width.

To calculate the appropriate pressure profiles inside the islands would require coupling a 3D equilibrium code to a transport code. Failing that, it is reasonable to take the pressure gradients flat in the islands. The equilibrium solutions thus obtained are, of course, fully self-consistent. Our code at present only deals with flat pressure profiles in the islands. We intend at a later time to incorporate the capability of dealing with pressure gradients in the islands to study their effect on island width.

Island formation leads to the appearance of regions of stochastic field lines near the separatrices. When islands get large enough to overlap, large regions of stochasticity appear. The pressure gradient is flattened in the stochastic regions, and the local current density vanishes everywhere in the stochastic regions.

3. The Algorithm

In this section we describe our nonvariational algorithm for solving the equilibrium equations, and we discuss our reasons for adopting it instead of an energy-minimizing algorithm.

The physics suggests an iterative algorithm. We begin with an initial guess for **B**. (For a stellarator this may be the vacuum field.) The pressure is taken to be constant along the field lines of **B**,

$$\mathbf{B}\cdot\nabla p = 0. \tag{4}$$

The diamagnetic current in this field is given by

$$\mathbf{j}_\perp = \mathbf{B} \times \nabla p \,/\, B^2. \tag{5}$$

We calculate the Pfirsch–Schlüter current from the magnetic differential equation

$$\mathbf{B}\cdot\nabla(j_\parallel / B) = -\nabla\cdot\mathbf{j}_\perp \tag{6}$$

The magnetic field is then updated using Ampere's law,

$$\nabla \times \mathbf{B} = \mathbf{j}. \tag{7}$$

The updated field is used, in turn, to recalculate the current. We continue cycling through this procedure until the changes in **B** from one iteration to the next become small. This algorithm was suggested early in the fusion program by Spitzer[5], and by Grad and Rubin[6]. The algorithm takes the plasma current as its fundamental quantity, in the sense that it is the quantity we calculate from the equilibrium equation, with the field expressed as a function of the current through Ampere's law. This contrasts with the energy-minimizing methods, where the magnetic field is the fundamental object that is calculated from the equilibrium equations, and the current may be calculated using Ampere's law.

In the axisymmetric case, the algorithm reduces to a conventional g-solver if we use the usual mixed representation for **B**,

$$\mathbf{B} = \nabla\Psi \times \nabla\varphi + g(\Psi)\nabla\varphi. \tag{8}$$

Ampere's law becomes

$$\mathbf{j} = \nabla \times \mathbf{B} = -\nabla\varphi\, \nabla^{*}\Psi + g'\nabla\Psi \times \nabla\varphi, \tag{9}$$

which reduces to the equation for Ψ,

$$\nabla^{*}\Psi = -j_{\varphi}, \tag{10}$$

where j_{φ} is the covariant φ component of $\mathbf{j}$. We determine j_{φ} from the equilibrium equation,

$$j_{\varphi} = R^{2}p'(\Psi) + gg'(\Psi). \tag{11}$$

Substituting Eq. (11) in Eq. (10), we obtain the conventional iterative scheme for solving the Grad-Shafranov equation:

$$\nabla^{*}\Psi_{n+1} = J_{\varphi}(\Psi_n) = R^2 p'(\Psi_n) + gg'(\Psi_n). \tag{12}$$

There are other possible algorithms that suggest themselves for calculating three-dimensional MHD equilibria with islands and stochastic regions. Perhaps the simplest conceptually would be to follow the time dependence of the plasma on a resistive time scale, watching the plasma relax to an equilibrium. This would have the advantage that issues related to detecting the presence of islands and stochastic regions, and modifying the the pressure and current profiles accordingly, would be handled automatically by the dynamics. Such a code would be very slow because of the need to resolve fast time scales. One would be led to methods which distort the fast time scales to accelerate convergence. This in turn leads naturally to the idea of neglecting all intermediate time scales to obtain a fast equilibrium solver. That has been the objective of three-dimensional MHD equilibrium codes.

The conventional approach that has been adopted in 3D equilibrium codes which assume good surfaces has been to take advantage of the variational formulation of the MHD equations. The algorithm evolves the magnetic field to a state of minimum energy, subject to the appropriate constraints. These methods have been extensively and successfully developed. Such a method could, in principle, be used without making any assumptions about the existence of flux surfaces. One possible way of implementing such a procedure, in fact, would be to make use of the "quasi-magnetic" coordinates introduced later in this paper. The numerical techniques developed in section 5 of this paper would, we believe, be useful also for dealing with islands and stochastic regions in the context of an energy-minimizing code. In deciding on an algorithm for our code, we seriously considered using an energy-minimizing method.

Our final decision to use the nonvariational method was dominated by our concern about the numerical difficulty of resolving the small energy associated with island formation in an energy-minimizing code. In studies of nonlinear MHD instabilities using the BETA code ([1], p. 15) energy differences associated with the MHD instabilities could only be resolved by extrapolating to zero grid spacing. In contrast, recent analytical calculations have demonstrated the capability of the nonvariational method for resolving islands in three-dimensional equilibria[4].

Historically, probably the greatest impediment to the adaptation of nonvariational methods to three dimensions has been uncertainty about their convergence properties in three dimensions. Energy-minimizing codes have the virtue that the basic algorithm is guaranteed to be convergent. No such guarantee is available for the nonvariational algorithms. There has accumulated considerable evidence, however, from analytical calculations, from axisymmetric codes, and from a recent mirror calculation that the nonvariational algorithm does generally have good convergence properties. The algorithm has been found to be a convenient method for obtaining analytical estimates of Shafranov shifts in three-dimensional equilibria[7,8]. Using a similar algorithm in the long thin limit to obtain 3D MHD equilibria for mirror machines, McNamara has claimed convergence much faster than that obtained by comparable variational codes[9].

The nonvariational algorithm has as an advantage that different coordinate systems can be used for solving the magnetic differential equations and for solving Ampere's law, so that optimal coordinate systems could be used for each. Because the boundary conditions come in through Ampere's law, a linear partial differential equation, the treatment of free boundaries would presumably be straightforward.

Recently, Betancourt and Garabedian have begun to modify the BETA code to remove the assumption that good flux surfaces exist[10]. It will be of great interest to compare the results of the two codes. After we began work on our code, we learned that Wobig was implementing the same algorithm in three dimensions using very different numerical methods[11] We look forward to comparison with the results of that code also.

4. Implementation of the Algorithm

In this section we outline our implementation of the algorithm presented in section 3. More details on the implementation will be presented in sections 5 and 6. In this section we also describe some of the mechanics of building the code.

Implementation of the nonvariational algorithm in three dimensions requires the development of two very different numerical capabilities. The first is to solve the magnetic differential equations that determine the pressure-driven current in a given field. The second is to solve Ampere's law for the magnetic field given the current.

To solve the magnetic differential equations, we construct a "quasi-magnetic" coordinate system (Ψ,Θ,Φ) such that

$$\mathbf{B} = \nabla\Psi \times \nabla\Theta + \iota\nabla\Phi \times \nabla\Psi + \mathbf{b}, \tag{13}$$

with b/B very small on the good (KAM) flux surfaces. This can be viewed as a mixed Eulerian-Lagrangian representation, with the coordinate system changing at each iteration to follow the good flux surfaces. On these surfaces, the magnetic differential equations reduce to algebraic equations. Elsewhere $\nabla p = 0$, so $\mathbf{j} = 0$ is determined without solving magnetic differential equations. Details will be discussed in section 5.

As we will show in section 6, the solution of Ampere's law in three dimensions can be reduced to the solution of a 3D Poisson equation with Neumann boundary conditions. To solve the three-dimensional Poisson equation, we Fourier decompose in θ and φ, and take second order finite differences in the radial direction. Solution of the Poisson equation then reduces to the inversion of a block tridiagonal matrix, for which we use a direct method. Further details are given in section 6.

The code has been written to run on the Cray computers at the U.S. Magnetic Fusion Energy Computer Center at Livermore. We do all our editing, precompiling, etc., locally on the Princeton Plasma Physics Lab's VAX's, shipping the Fortran source code to Livermore over the MFE network. Although this sometimes makes for an awkward debugging cycle, we find the inconvenience more than compensated for by the increased availability of software tools on the local VAX's. In particular, we do our editing using a screen editor on the VAX.

The code is written in Ratfor, and run through a Ratfor[12] preprocessor on the VAX to produce the Fortran source code. Ratfor is an extension of Fortran with modern control flow statements, such as "while" and "repeat-until", that facilitate structured programming. It also has a somewhat modified format from Fortran's to improve the readability of code.

A macro preprocessor on the VAX allows us to abbreviate some commonly used sequences of commands. For example, we have an

"ABRTIF" (abort if) macro that checks for problems at various points in the code.

Other Unix-like software tools[12] available on the VAX have aided the modular construction of the code.

The code uses heap allocation and deallocation to make efficient use of memory. This is also convenient because it allows us to specify the dimensions of our arrays as input parameters, and to dynamically change dimensions of arrays within a loop.

We have found that the software tools on the VAX have had a major impact on our efficiency in developing the code. We feel that this is an insufficiently appreciated aspect of developing large codes that can have as great an impact as hardware development.

5. "Quasi-Magnetic" Coordinates for Magnetic Fields with Islands and Stochastic Regions

As described in the previous section, we solve our magnetic differential equations by transforming to "quasi-magnetic" coordinates, in which **B** takes the form shown in Eq. (13). In this section we describe our algorithm for transforming the coordinate system, demonstrate how the algorithm works on some model fields with islands and stochastic regions, and explain how we use the quasi-magnetic coordinate system to solve our magnetic differential equations. We also describe the numerical errors arising in this part of the code, and how they are controlled.

For a field line that covers a good flux surface, it should not be surprising that the Fourier decomposition of the coordinates along the field line contains sufficient information to determine the flux surface. An algorithm for constructing magnetic coordinates in the surface from this given Fourier information was introduced for 3D transport applications by Boozer[13]. Some aspects of that algorithm are incompletely automated. Althougn not an issue for applications to transport, it is not adequate for our application in an equilibrium code. We have adapted Boozer's ideas to develop the algorithm we require for constructing magnetic coordinates on our good flux surfaces.

If we apply this algorithm in an island or a stochastic region, we find that the field line deviates from the surface defined by the Fourier components of the coordinates. This provides a sensitive diagnostic which tells us whether we are on a good flux surface. In the islands and the stochastic regions we choose a set of coordinates which smoothly interpolate between the coordinates in the good regions. These coordinates are of course not uniquely defined.

We first consider the good flux surfaces. Any single-valued function on a good flux surface can be Fourier decomposed in magnetic coordinates θ and ϕ:

$$f(\theta,\phi) = \sum_{n,m} f_{nm} \exp[\ i\ (\ n\phi - m\theta\ )\ ]. \qquad (14)$$

On a single field line the magnetic coordinates satisfy $\theta = \iota\,\phi$ (see Eq. (2)). Equation (14) can be written

$$f = \sum_{n,m} f_{nm} \exp[\ i\ (n - \iota\, m)\ \phi\ ]. \qquad (15)$$

This shows that f_{nm} can be determined by an FFT along a single field line, if we can identify n and m for the discrete spectral lines in the 1D spectrum. To follow the field lines, we integrate the pair of ordinary differential equations,

$$d\theta / d\phi = \mathbf{B}\cdot\nabla\theta / \mathbf{B}\cdot\nabla\phi, \quad d\rho / d\phi = \mathbf{B}\cdot\nabla\rho / \mathbf{B}\cdot\nabla\phi.$$

Figure 1 shows a typical spectrum we have obtained by an FFT along a field line. Instead of discrete lines, the numerical calculation gives spectral peaks whose widths are related to the distance over which the field lines have been followed.

To determine the magnetic coordinates implicitly, we can Fourier decompose

$$(x,y,z) = \mathbf{x} = \sum_{n,m} \mathbf{x}_{nm} \exp[\ i\ (n - \iota\, m)\ \phi\ ]. \qquad (16)$$

This gives us our coordinate transformation, once we know how to identify the spectral lines in the one-dimensional spectrum.

We proceed by first obtaining an accurate solution for the rotational transform. The mode numbers m and n associated with each of the spectral lines in the one-dimensional spectrum, Eq. (15), can then be identified directly. Because of the expense of following field lines, we need an efficient algorithm for calculating ι. The conventional method of numerically calculating $\lim(\theta/\varphi)$ would require many orbits to calculate ι to the required accuracy.

To find the rotational transform efficiently, we use an iterative procedure that chooses ι at each step to minimize

$$F(\iota) \equiv \sum_j \{ \theta(\Phi_j) - \iota\Phi_j - \sum_{n,m} \theta_{nm} \sin[(n - \iota m)\Phi_j] \}^2$$

with respect to ι, where $\Phi_j = j\,\Delta\Phi$ are the points sampled on the field line, and the geometric poloidal angle $\theta(\Phi_j)$ is determined by following the field line. We take Φ to be the geometric toroidal angle, φ. The inner sum is over the Fourier components of θ. For the first step we take all $\theta_{nm} = 0$ and minimize F to get an initial ι. At each subsequent step we evaluate θ_{nm} by a fast Fourier transform of $\theta - \iota\Phi$, using the current ι. F is then minimized to update ι. This algorithm makes efficient use of the information obtained in following field lines, and thus minimizes the expensive field line following. It has been found to converge rapidly in practice.

The functions we are Fourier decomposing along the field lines are not generally periodic. To deal with the nonperiodicity we use the standard technique[14] of multiplying by an appropriate window function. We use a Gaussian window, $\exp[-(\varphi/\varphi_g)^2]$, which permits analytical error estimates.

In determining the Fourier coefficients by Fast Fourier Transform, we keep the numerical errors within a prescribed tolerance. These numerical errors become a particular issue if the surface we are on is near a rational surface, as discussed below.

Numerical inaccuracy arises from the finite integration length and from spectral leakage. The relative error in each Fourier coefficient due to the finite integration length, φ_f, is

$$(1/\sqrt{\pi})\,(\varphi_g/\varphi_f)\,\exp[-(\varphi_f/\varphi_g)^2]. \tag{17}$$

The error due to spectral leakage from a second mode of amplitude a is,

$$a\left[\frac{1}{\sqrt{\pi}}\,\frac{\cos(\Delta k\,\varphi_f)\,\exp[-(\varphi_f/\varphi_g)^2]\,(\varphi_f/\varphi_g)}{(\varphi_f/\varphi_g)^2+(\Delta k\,\varphi_g)} + \exp[-(\Delta k\,\varphi_g)^2]\right]. \tag{18}$$

where Δk is the distance in k-space between modes. This arises from interference due to the nonorthogonality of the Fourier modes. The spectral leakage from a set of modes is linearly additive. The code chooses φ_g and φ_f to keep the total error within a prescribed tolerance.

As we get close to a rational surface, Δk decreases, requiring increased φ_f. This problem only arises near those surfaces where $\iota = n/m$, with $0 \le m \le 2M$ and $-2N \le n \le 2N$, and where M and N determine the modes included in the analysis. We cannot apply the method right on one of these rational surfaces. In practice, where the island widths are small compared to the radial grid spacing we determine the Fourier coefficients on the coordinate surface closest to one of these rational surfaces by interpolating the coefficients on the neighboring surfaces. The Δk which determines our required integration length is then determined by the local shear and by the radial grid spacing. As we increase the number of radial grid points, Δk decreases, requiring increased φ_f.

In the presence of large islands, we also need to worry about the distortion of the surfaces near the separatrix. As we get close to a separatrix, the surfaces require increasing numbers of Fourier modes for an accurate representation. We cannot apply the method right at a separatrix. We demand that our required tolerances be maintained on all good surfaces that are more than one radial grid spacing from a separatrix. This requires the number of Fourier modes to increase as the number of radial grid points increases. The severity of this constraint is reduced by the presence of stochastic regions in the neighborhood of the separatrices.

Now we turn to the issue of dealing with islands and stochastic regions. Those regions are treated differently from the regions where there are good surfaces. The coordinate system is calculated there by interpolating between the regions of good surfaces. The gradient of the pressure will be set to zero there. For these purposes we need a diagnostic for distinguishing the good surfaces from the islands and stochastic regions.

We proceed by applying our algorithm for calculating flux surfaces everywhere. We then calculate the mean square deviation of the field lines from the reconstructed flux surfaces:

$$\frac{1}{L} \sum_j \{ \mathbf{x}(\varphi_j) - \sum_{n,m} \mathbf{x}_{nm} \exp[i(n - \ell m) \varphi_j] \}^2 , \qquad (19)$$

where L is the number of points sampled along a field line. On the good surfaces this quantity is small compared to the radial grid spacing. In the islands and stochastic regions it is of order one. The coordinates in the islands and stochastic regions are recalculated by interpolating the $\mathbf{x}_{nm}$.

To test our algorithm for constructing quasi-magnetic coordinates, we use the (manifestly divergence-free) model field:

$$\mathbf{B} = \nabla\Psi_t \times \nabla\theta + \nabla\Phi \times \nabla\Psi_p ,$$

with $\Psi_t = r^2/2$ and

$$\Psi_p = \iota_0 \Psi_t + \iota_1 \Psi_t^2 - \varepsilon_1 r^2 \cos(2\theta - \Phi) - \varepsilon_2 r^3 \cos(3\theta - \Phi).$$

Note that Ψ_p defines a (time-dependent) Hamiltonian for the field line orbits, with canonical coordinates Ψ_t, θ, and with Φ playing the role of time. In the examples shown in figures 2 and 3, we have taken $\iota_0 = 0.29$, $\iota_1 = 0.38$. As ε_1 and ε_2 are increased from zero, we get islands and stochastic regions. For the examples shown the code was run with 10 radial coordinate surfaces and with $0 \leq m \leq 6$, $-3 \leq n \leq 3$. The initial coordinates for the 10 field lines followed were chosen by dividing the interval between the magnetic axis and the right-hand boundary on the midplane into 10 equal segments. In Figure 3 the 7th and 9th lines out from the axis lie on nearby island surfaces, giving a broadened appearance to these surfaces in the figure. It is clear from the figures that the radial coordinate does track the good flux surfaces closely and smoothly interpolates elsewhere. Note particularly in Fig. 3 that the second surface from the axis is very close to a separatrix, and is therefore strongly distorted. The mean square deviation of the field line from the corresponding coordinate surface is nevertheless less than 0.002. The deviation is caused by the limited number of Fourier modes retained.

Our purpose in transforming to quasi-magnetic coordinates is to solve the magnetic differential equation for the current. In a flux coordinate system with straight field lines, magnetic differential equations reduce to algebraic equations. The solution for the plasma current with a given $p'(\Psi)$ (and zero net current) is:

$$\mathbf{j} = dp/d\Psi \left[\nabla\Phi \times \nabla\Psi \ + \sum_{n,m}{}' \frac{\mathcal{J}_{nm}}{(n - \iota m)} \cos(n\Phi - m\Theta)\ \nabla\Psi \times (m\nabla\Theta - n\nabla\Phi) \right], \quad (20)$$

where the $\mathcal{J}_{nm}$ are the Fourier components of the Jacobian,

$\mathcal{J} = 1 / (\mathbf{B}\cdot\nabla\Phi)$. We have taken Φ to be the uniform toroidal angle, so $\mathbf{B}\cdot\nabla\Phi$ is known, and we can use our algorithm to find its Fourier decomposition on the good flux surfaces. Since we take $\nabla p = 0$ in the islands and the stochastic regions, $\mathbf{j} = 0$ there.

6. The Solution of Ampere's Law in Three Dimensions

In the previous section we described how we solve for the current. In this section we describe the solution of Ampere's law to update the magnetic field, given the current.

The solution of Ampere's law in three dimensions can be reduced to the solution of a three-dimensional Poisson equation as follows. First we write the magnetic field in contravariant form. Then we set the components of its curl equal to the components of $\mathbf{j}$ given by Eq. (20). The resulting equations can be integrated with respect to the radial coordinate to give a solution $\mathbf{h}$ such that

$$\nabla \times \mathbf{h} = \mathbf{j}, \tag{21a}$$

but

$$\nabla \cdot \mathbf{h} \neq 0 . \tag{21b}$$

The general solution of $\mathbf{B}$ is

$$\mathbf{B} = \mathbf{h} + \nabla u, \tag{22a}$$

where $\nabla \cdot \mathbf{B} = 0$ implies

$$\nabla^2 u = - \nabla \cdot \mathbf{h}. \tag{22b}$$

The boundary conditions for this equation are determined by those on $\mathbf{B}$. In particular, we assume a fixed outer flux surface, which gives a Neumann condition on u:

$$(\mathbf{h} + \nabla u) \cdot \nabla\Psi = \mathbf{B} \cdot \nabla\Psi = 0$$

In performing the radial integrations to solve for $\mathbf{h}$, we choose the constant of integration so that only the radial contravariant component is nonvanishing for the $m \neq 0$ or $n \neq 0$ Fourier components,

$$\mathbf{h} = \nabla\Phi \int_{\Psi}^{\Psi_e} \mathcal{J}_0\, p'(\Psi)\, d\Psi \;+\; \nabla\Psi\, p'(\Psi) \textstyle\sum'_{nm} \sin(n\Phi - m\Theta)\, \mathcal{J}_{nm} / (n - \iota m). \tag{23}$$

This solution for $\mathbf{h}$ was suggested by Boozer[8].

In the previous section we solved for $\mathbf{j}$ in a numerically specified set of quasi-magnetic coordinates. We have obtained the solution for $\mathbf{h}$ in the same coordinates. At this point, we could transform to a new coordinate system for the solution of the Poisson equation. Presently, the Poisson equation is solved in the same quasi-magnetic coordinate system used to solve for **j**.

In general coordinates, Poisson's equation takes the form:

$$\partial_i(\sqrt{g}\, g^{ij}\, \partial_j u) = -\partial_i(\sqrt{g}\, g^{ij}\, h_j)\,. \tag{24}$$

The coordinates are numerically specified through the Fourier coefficients of $\mathbf{x}(\rho,\theta,\varphi)$. We calculate $\sqrt{g}\, g^{ij}$ numerically from the equations

$$\sqrt{g}\, g^{ij} = \sqrt{g}^{\,-1}\, g_{ij}^{\,-1}\,, \tag{25a}$$

$$g_{ij} = (\partial\mathbf{x}/\partial\xi^i)\cdot(\partial\mathbf{x}/\partial\xi^j)\,. \tag{25b}$$

The angular derivatives are evaluated directly in Fourier space, and the radial derivatives are evaluated on the mesh points using centered finite differences.

We solve the three-dimensional Poisson equation for u using a Fourier representation in θ and φ, and 2nd order finite differences in the radial variable ρ. The Poisson equation becomes a matrix equation, with a coefficient matrix which is block tridiagonal. We invert this matrix with Gaussian elimination, using a routine due to Hindmarsh[15]. This routine requires only one block row in memory at a time and is particularly useful for large matrices.

We have made the Poisson solver fully spectral to avoid potential problems due to aliasing. Multiple products are evaluated by repeated pairwise convolution. For Eq. (25a) we use the exact spectral inverse of $\mathcal{J}$

To increase the accuracy of the radial derivatives, we make use of our *a priori* knowledge of the behavior of the Fourier coefficients near the origin. For example, we know that u_{mn} goes like ρ^m near the origin. We pull out the ρ^m factor in each Fourier term up to m = maxexp, where maxexp is specified as an input parameter to the code. For example, if m < maxexp, we evaluate

$$\partial_\rho u_{mn} = m u_{mn}/\rho + \rho^m \partial_\rho (u_{mn}/\rho^m). \qquad (26)$$

Note that some terms are otherwise singular near the origin: $\mathcal{J} g^{\theta\theta} \sim \rho^{-1}$. For m > maxexp, we pull out a ρ^{maxexp}. In the absence of the maxexp cutoff, the discrete Poisson equation becomes ill posed because of matrix elements which are smaller than the machine precision.

We have found that the numerical stability of the code is sensitive to the form of the radial finite differencing used in the Poisson equation. In our initial version of the code we expanded the Poisson equation before differencing:

$$(\partial_i \mathcal{J} g^{ij}) \partial_j u + \mathcal{J} g^{ij} \partial_i \partial_j u = -\partial_i (\mathcal{J} g^{ij}) h_j - \mathcal{J} g^{ij} \partial_i h_j. \qquad (27)$$

This is inconsistent with conservative differencing. However, the flux is not a conserved quantity in our algorithm. Also, the expanded form allows us to make optimal use of our knowledge of the ρ^m scaling near the origin and is therefore more accurate there. This scheme unfortunately proved to be numerically unstable. The instability appears to enter through cancellations between large terms in satisfying the boundary condition. Short wavelength radial oscillations develop in the Fourier coefficients of the solution at the outer boundary, and they grow in time.

We have modified the code to use conservative differencing (aside from the ρ^m factors). This has cured the instability. The values of u are determined on a grid shifted radially by half a grid spacing relative to the coordinate grid. Shifting the u grid in this way provides increased accuracy in imposing the Neumann boundary condition. The derivatives of u and the metric elements are evaluated on the coordinate grid. Angular derivatives of u are evaluated there by interpolating between the u grid points. The covariant components of **B** are then constructed on the coordinate grid. The Poisson operator (i.e., $\nabla \cdot \mathbf{B}$) is evaluated on the u grid. Interpolation is again necessary to evaluate the angular derivatives.

7. A Three-Dimensional Equilibrium

To illustrate the properties of the code, we consider a 5 period, $l=2$ stellarator with an aspect ratio of 10 and a transform of approximately 0.14 at the center. A relatively low value of the rotational transform was chosen so that we could study the convergence properties of the code without the added complication of resonant surfaces. The assumed pressure profile is $p = p_0 (1 - \rho^2)^2$, with a beta of 6×10^{-4}. The corresponding Shafranov shift is about 10%. We ran the code for this case with 49 Fourier modes ($0 \le m \le 6$, $-3 \le n \le 3$) and 20 radial grid surfaces.

We used the Bessel function solution for the vacuum field in a cylinder as an initial guess for the field. The code preserves the shape of the outer flux surface and the toroidal line integral of the toroidal field at the outer surface (i.e., the current through the hole of the torus). It enforces zero net current through each flux surface.

For zero pressure, the code gives the toroidal vacuum solution in one iteration.

Figures (4a) and (4b) show the coordinate grid initially and after the plasma has reached equilibrium. The Shafranov shift of the flux surfaces is apparent in Fig. (4b). The constant θ surfaces show the distortion of the poloidal angle with beta.

Figure 1 shows the Fourier decomposition of the Cartesian x coordinate along a field line in the equilibrium solution. The field line lies in the outer flux surface. The initial field has nonzero Fourier coefficients only at $n = \iota m$, corresponding to the major peaks in Fig.

1. The mode coupling due to the presence of toroidicity broadens these peaks, giving finite amplitudes for all n and m. In Fig. 1 the peaks have been further broadened by the nonlinear coupling in the presence of finite beta.

Table 1 shows the values of three quantities that we monitor as the code progresses. The first is the maximum residual force normalized to the peak pressure:

$$F_{max} = \max(|\mathbf{j} \times \mathbf{B} - \nabla p|) / p_0 .$$

The maximum relative correction to the Fourier coefficients of the coordinates and Jacobian is labeled corr. The relative shift in the magnetic axis from the previous step is labeled xax. This quantity is normalized relative to the distance of the magnetic axis from the right-hand boundary at the current step.

The 0'th iteration corresponds to the initial field. After the third iteration the residual force stops decreasing, probably due to radial discretization errors. The relative corrections and the magnetic axis shift continue to decrease.

This run took 2 minutes and 20 seconds on the Cray 1.

8. Conclusion

We have described a code that is capable of solving for three-dimensional MHD equilibria without assuming the existence of good flux surfaces. The code uses an unconventional nonvariational algorithm. It updates the magnetic field at each step by calculating the pressure-driven current in the present field and then solving Ampere's law. We have shown how the magnetic differential equations for the pressure-driven current can be solved even in the presence of magnetic islands and stochastic field lines. The solution of Ampere's law is reduced to the solution of a three-dimensional Poisson equation. We have developed an algorithm for solving the three-dimensional Poisson equation in arbitrary numerically specified coordinates. We have also dealt with the issue of how to do the finite differencing for the Poisson solver so that the code is numerically stable. Preliminary runs for some stellarator equilibria having good flux surfaces indicate that the convergence properties of the code will be quite favorable. Details of the Poisson solver[16] and of the magnetic differential equation solver[17] will be presented in forthcoming publications.

We have not yet made a systematic study of the convergence properties of our code for a large class of three-dimensional equilibria, nor have we studied the convergence properties in the presence of large islands and stochastic regions. Those subjects will remain for a future paper.

Acknowledgments

We have benefited greatly from numerous discussions with

Allen Boozer and Don Monticello.

This work was supported by U.S. Dept. of Energy Contract No. DE-AC02-76CHO3073.

References

[1] F. Bauer, O. Betancourt, and P. Garabedian, Magnetohydrodynamic Equilibrium and Stability of Stellarators (Springer-Verlag, New York, 1984).

[2] S.P. Hirshman and D.K. Lee, Comput. Phys. Comm. 39 (1986) 161; S.P. Hirshman and P. Merkel, Comput. Phys. Comm., to appear.

[3] U. Schwenn, Comput. Phys. Comm. 31 (1984) 167.

[4] A.H. Reiman and A.H. Boozer, Phys. Fluids 27 (1984) 2446.

[5] L. Spitzer, Phys. Fluids 1 (1958) 253.

[6] H. Grad and H. Rubin, in: Proceedings of the Second International Conference on the Peaceful Uses of Atomic Energy (United Nations, Geneva, 1958), Vol. 31, p. 190.

[7] L. S. Solov'ev and V. D. Shafranov, in: Reviews of Plasma Physics, ed. M. A. Leontovich (Consultants Bureau, New York, 1970) p. 117.

[8] A.H. Boozer, Phys. Fluids 27 (1984) 2110.

[9] B. McNamara in: Proceedings of the U.S.-Japan Workshop on 3D MHD Studies (Oak Ridge, Tennessee, 1984) p. 113.

[10] O. Betancourt and P. Garabedian, 1986 Sherwood Theory Conference (New York, April 14-16), paper 2D-23.

[11] J. Kisslinger and H. Wobig, 12th European Conference on Controlled Fusion and Plasma Physics (Budapest, Sept. 1985),

Contributed Papers, Part 1, p. 453.

[12] B.W. Kernighan and P.J. Plauger, Software Tools (Addison-Wesley, Reading, Mass., 1976).

[13] A.H. Boozer, Phys. Fluids 25 (1982) 520.

[14] F.J. Harris, Proceedings of the IEEE 66 (1978) 51.

[15] A.C. Hindmarsh, Lawrence Livermore National Laboratory Report UCID-30150 (1977).

[16] H. Greenside and A. Reiman, to be published.

[17] A. Reiman and H. Greenside, to be published.

Figure Captions

Figure 1. Numerical Fourier transform of the Cartesian x coordinate with respect to the geometric toroidal angle φ along a magnetic field line. The magnetic field is that of the stellarator equilibrium described in section 7. The field line lies in the outer flux surface.

Figure 2. (a) Poincaré plot at $\varphi = 0$ for the model field described in the text with $\varepsilon_1 = 0.0015$ and $\varepsilon_2 = 0.005$. (b) Coordinate grid at $\varphi = 0$ with the same field.

Figure 3. Same as Fig. 2, except that $\varepsilon_1 = 0.005$ and $\varepsilon_2 = 0.015$.

Figure 4. Coordinate grid at $\varphi = 0$ for the (a) initial and (b) equilibrium stellarator fields. The radial coordinate surfaces coincide with flux surfaces.

iteration	F_{max}	corr	xax
0	0.21		
1	0.12	0.11	0.03
2	2.5×10^{-3}	4.5×10^{-2}	0.07
3	5.4×10^{-4}	1.0×10^{-3}	9.3×10^{-4}

Table 1. Three quantities monitored during convergence of the stellarator equilibrium.

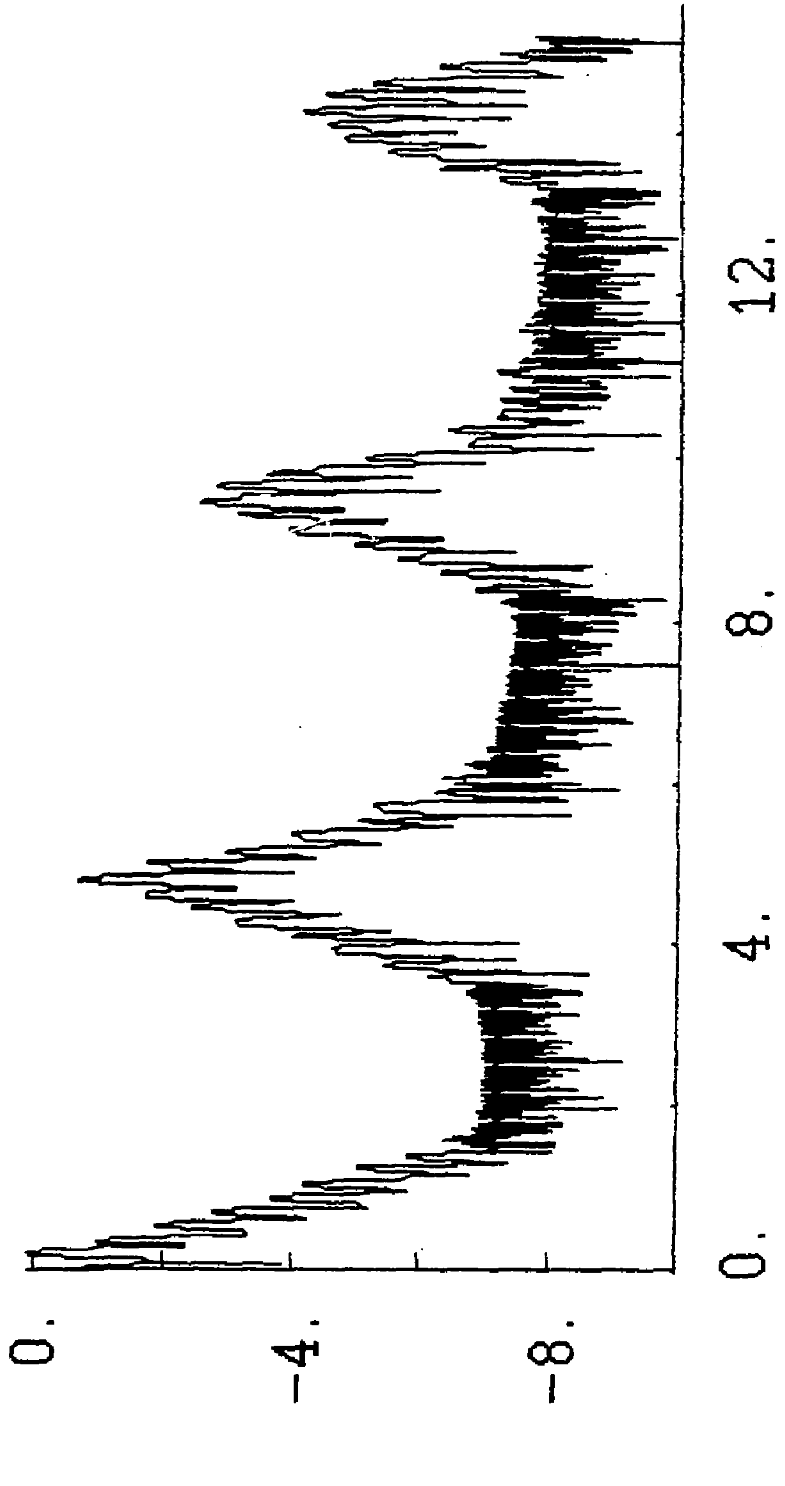

Fig. 1

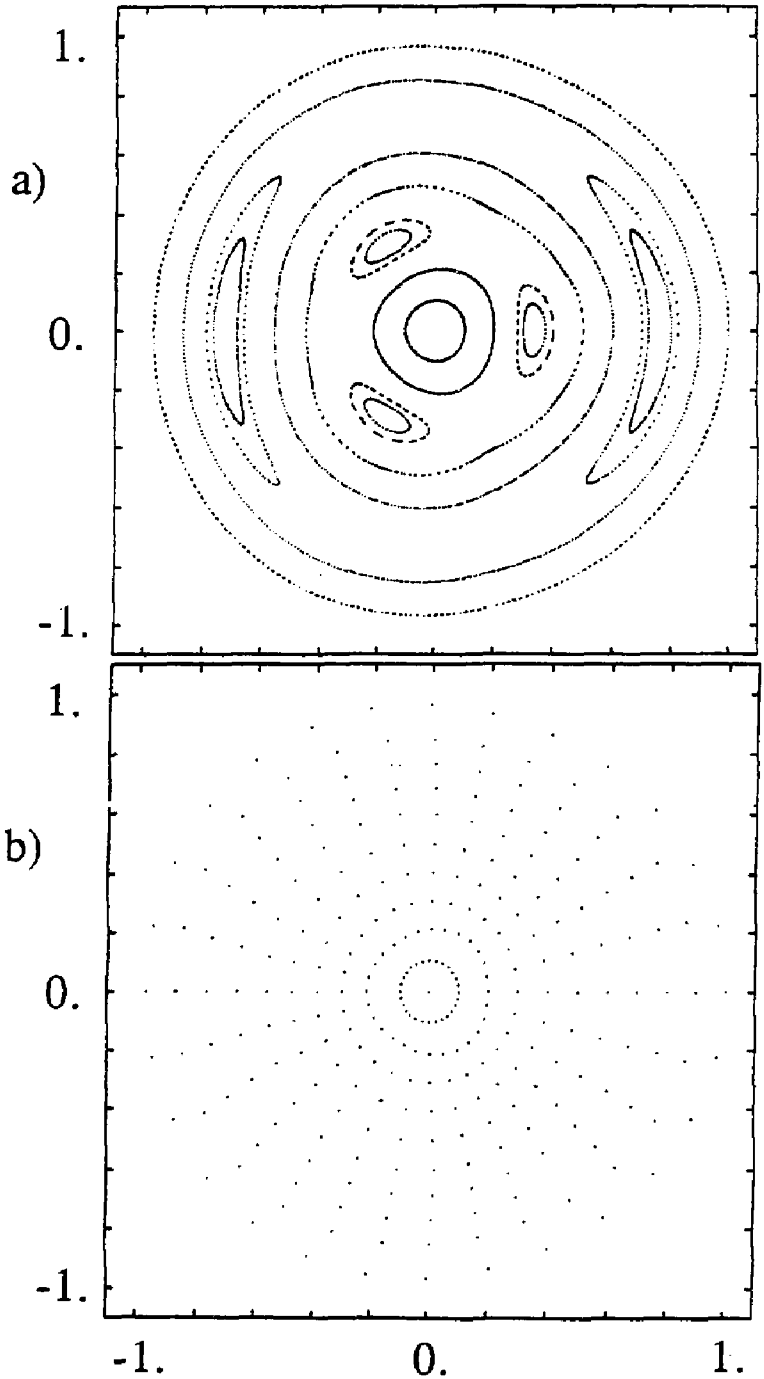

Fig. 2

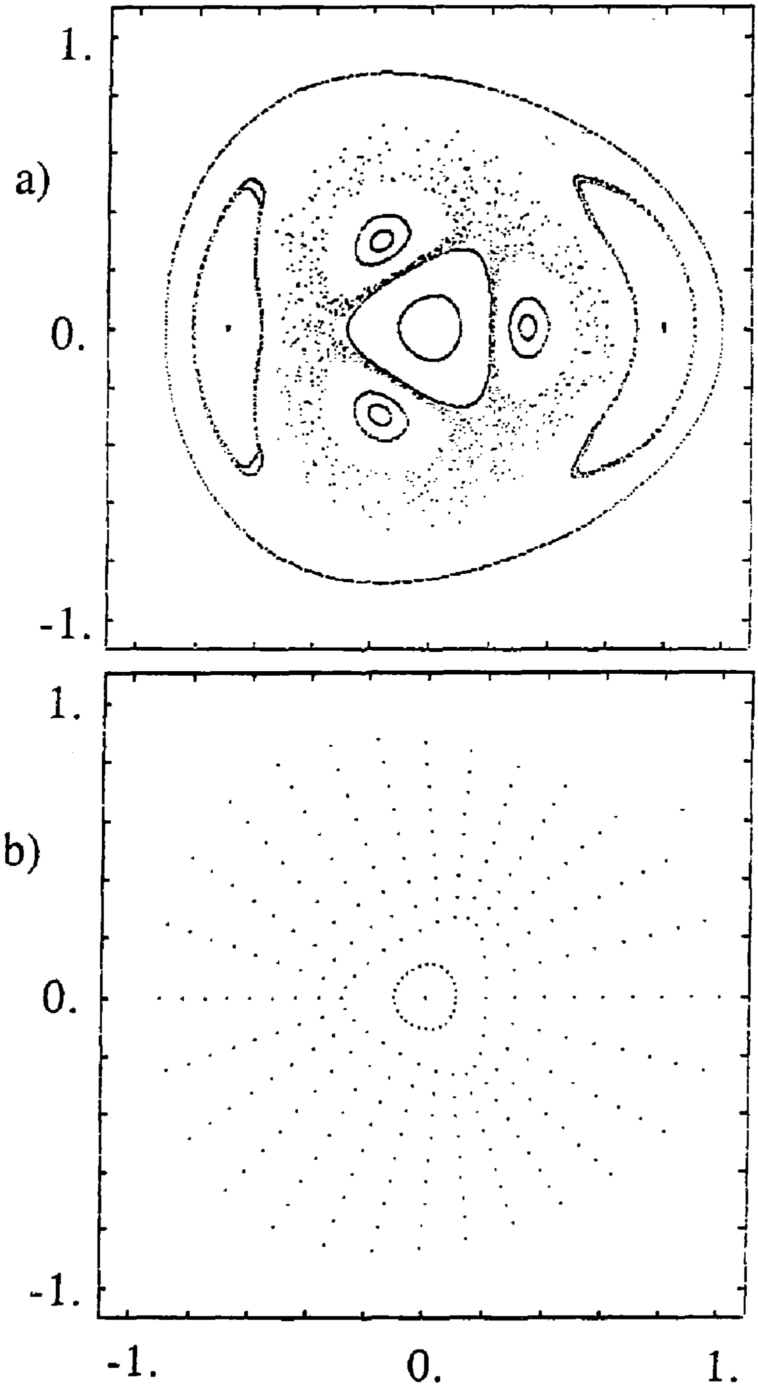

Fig. 3

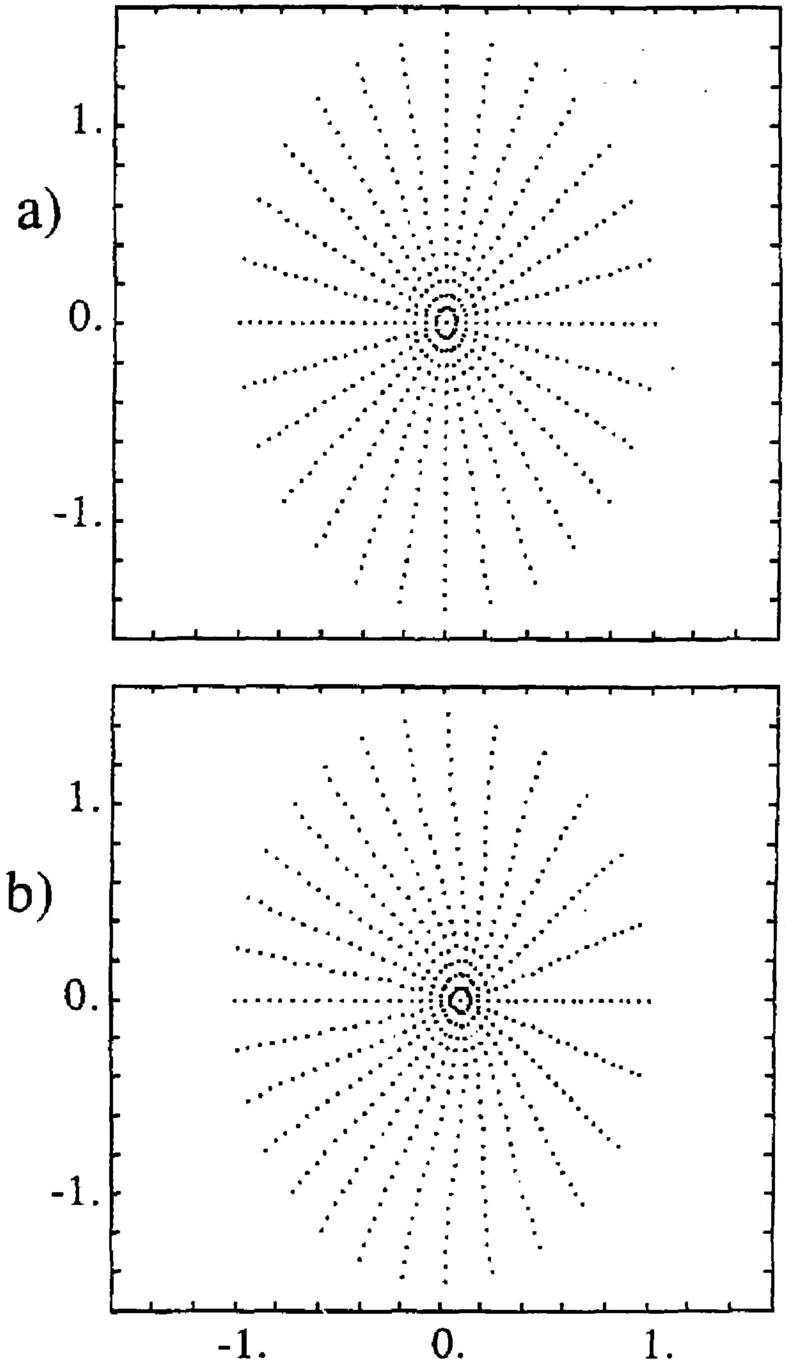

Fig. 4

EXTERNAL DISTRIBUTION IN ADDITION TO UC-20

Plasma Res Lab, Austra Nat'l Univ, AUSTRALIA
Dr. Frank J. Paoloni, Univ of Wollongong, AUSTRALIA
Prof. I.R. Jones, Flinders Univ., AUSTRALIA
Prof. M.H. Brennan, Univ Sydney, AUSTRALIA
Prof. F. Cap, Inst Theo Phys, AUSTRIA
M. Goossens, Astronomisch Instituut, BELGIUM
Prof. R. Boucique, Laboratorium voor Natuurkunde, BELGIUM
Dr. D. Palumbo, Dg XII Fusion Prog, BELGIUM
Ecole Royale Militaire, Lab de Phys Plasmas, BELGIUM
Dr. P.H. Sakanaka, Univ Estadual, BRAZIL
Lib. & Doc. Div., Instituto de Pesquisas Espaciais, BRAZIL
Dr. C.R. James, Univ of Alberta, CANADA
Prof. J. Teichmann, Univ of Montreal, CANADA
Dr. H.M. Skarsgard, Univ of Saskatchewan, CANADA
Prof. S.R. Sreenivasan, University of Calgary, CANADA
Prof. Tudor W. Johnston, INRS-Energie, CANADA
Dr. Hannes Barnard, Univ British Columbia, CANADA
Dr. M.P. Bachynski, MPB Technologies, Inc., CANADA
Chalk River, Nucl Lab, CANADA
Zhengwu Li, SW Inst Physics, CHINA
Library, Tsing Hua University, CHINA
Librarian, Institute of Physics, CHINA
Inst Plasma Phys, Academia Sinica, CHINA
Dr. Peter Lukac, Komenskeho Univ, CZECHOSLOVAKIA
The Librarian, Culham Laboratory, ENGLAND
Prof. Schatzman, Observatoire de Nice, FRANCE
J. Radet, CEN-BP6, FRANCE
JET Reading Room, JET Joint Undertaking, ENGLAND
AM Dupas Library, AM Dupas Library, FRANCE
Dr. Tom Mual, Academy Bibliographic, HONG KONG
Preprint Library, Cent Res Inst Phys, HUNGARY
Dr. R.K. Chhajlani, Vikram Univ, INDIA
Dr. B. Dasgupta, Saha Inst, INDIA
Dr. P. Kaw, Physical Research Lab, INDIA
Dr. Phillip Rosenau, Israel Inst Tech, ISRAEL
Prof. S. Cuperman, Tel Aviv University, ISRAEL
Prof. G. Rostagni, Univ Di Padova, ITALY
Librarian, Int'l Ctr Theo Phys, ITALY
Miss Clelia De Palo, Assoc EURATOM-ENEA, ITALY
Biblioteca, del CNR EURATOM, ITALY
Dr. H. Yamato, Toshiba Res & Dev, JAPAN
Direc. Dept. Lg. Tokamak Dev. JAERI, JAPAN
Prof. Nobuyuki Inoue, University of Tokyo, JAPAN
Research Info Center, Nagoya University, JAPAN
Prof. Kyoji Nishikawa, Univ of Hiroshima, JAPAN
Prof. Sigeru Mori, JAERI, JAPAN
Prof. S. Tanaka, Kyoto University, JAPAN
Library, Kyoto University, JAPAN
Prof. Ichiro Kawakami, Nihon Univ, JAPAN
Prof. Satoshi Itoh, Kyushu University, JAPAN
Dr. D.I. Choi, Adv. Inst Sci & Tech, KOREA
Tech Info Division, KAERI, KOREA
Bibliotheek, Fom-Inst Voor Plasma, NETHERLANDS
Prof. B.S. Liley, University of Waikato, NEW ZEALAND
Prof. J.A.C. Cabral, Inst Superior Tecn, PORTUGAL
Dr. Octavian Petrus, A.I CUZA University, ROMANIA
Prof. M.A. Hellberg, University of Natal, SO AFRICA
Dr. Johan de Villiers, Plasma Physics, Nucor, SO AFRICA
Fusion Div. Library, JEN, SPAIN
Prof. Hans Wilhelmson, Chalmers Univ Tech, SWEDEN
Dr. Lennart Stenflo, University of UMEA, SWEDEN
Library, Royal Inst Tech, SWEDEN
Centre de Recherchesen, Ecole Polytech Fed, SWITZERLAND
Dr. V.T. Tolok, Kharkov Phys Tech Ins, USSR
Dr. D.D. Ryutov, Siberian Acad Sci, USSR
Dr. G.A. Eliseev, Kurchatov Institute, USSR
Dr. V.A. Glukhikh, Inst Electro-Physical, USSR
Institute Gen. Physics, USSR
Prof. T.J.M. Boyd, Univ College N Wales, WALES
Dr. K. Schindler, Ruhr Universitat, W. GERMANY
ASDEX Reading Rm, IPP/Max-Planck-Institut für Plasmaphysik, F.R.G.
Nuclear Res Estab, Julich Ltd, W. GERMANY
Librarian, Max-Planck Institut, W. GERMANY
Bibliothek, Inst Plasmaforschung, W. GERMANY
Prof. R.K. Janev, Inst Phys, YUGOSLAVIA