Convergence Properties of a Nonvariational 3D MHD Equilibrium Code

H. S. GREENSIDE,* A. H. REIMAN, AND A. SALAS†

Plasma Physics Laboratory, Princeton University,
Princeton, New Jersey 08544-0451

Received July 17, 1987; revised April 28, 1988

Details are presented of the PIES code, which uses a nonvariational algorithm for calculating fully three-dimensional MHD equilibria. The convergence properties of the code are studied for several axisymmetric and nonaxisymmetric finite β equilibria that have magnetic surfaces.

I. INTRODUCTION

Computer codes for calculating toroidal magnetohydrodynamic (MHD) equilibria play an important role in the design of plasma fusion machines and in the interpretation of experimental data. Desirable goals for such codes are the ability to simulate 3-dimensional (3D) equilibria efficiently (since the parameter space for optimizing fusion machines is high-dimensional) and to treat magnetic fields with stochastic field lines and magnetic islands. The latter goal is especially important since symmetries that guarantee the existence of magnetic surfaces everywhere may be lacking (stellarators) or may be spontaneously broken (tearing modes in tokamaks).

Impressive progress has been made in developing accurate and efficient 3-dimensional equilibrium solvers [1–7] but progress towards the second goal has been much less satisfactory. The main difficulty lies in developing numerical methods that allow magnetic fields to change topology. Existing 3-dimensional equilibrium codes either assume the existence of magnetic surfaces [1–5] or impose flux conservation while evolving the magnetic field towards a minimum on an Eulerian grid [6, 7]. These restrictions do not allow changes in the topology of the initial magnetic field and have prevented useful simulations of equilibria with islands to date.

These difficulties suggest that it would be fruitful to explore new approaches that might lead to efficient codes that can also treat magnetic islands and stochastic field lines. New approaches would also be interesting in their own right.

In a recent paper by two of the authors [8], preliminary details were given of a

[*] Permanent address: Department of Computer Science, Duke University, Durham, NC 27706.

[†] Permanent address: Departmento de Fisica, E.T.S. de Ingenieros Industriales, Universidad Politécnica de Madrid, José Gutiérrez Abascal, 2, 28006 Madrid, Spain.

0021-9991/89 $3.00

new 3D MHD equilibrium code (called PIES, for the Princeton Iterative Equilibrium Solver) that offered certain potential advantages in resolving these difficulties. Unlike existing variational codes, the PIES code implemented a nonvariational algorithm similar to the one first proposed by Spitzer [9] and by Grad and Rubin [10] in the 1950's. MHD equilibria are then determined as fixed points of a nonlinear iteration scheme rather than as minima of the MHD functional. As pointed out in several recent papers [11, 12, 8], this offers the possibility of treating islands and stochastic field lines in an elegant manner without sacrificing efficiency. An additional benefit is a natural way of treating zero net toroidal current, which is necessary for stellarators.

In the present paper, we give a more complete description of the PIES code, and a careful study of its convergence properties. Obvious important questions are whether nonvariational codes converge at all and how convergence depends on numerical and physical parameters. Since nonvariational algorithms have not been studied previously in 3D equilibrium codes, we answer this question first for the case in which magnetic surfaces exist everywhere. This gives a foundation for future studies with applications to equilibria with magnetic islands and to tearing modes in tokamaks [13].

A key result given below is that the PIES code does converge for representative tokamak and stellarator equilibria with magnetic surfaces and that the convergence is both rapid (few iterations needed) and efficient (moderate CPU time needed per iteration). Nonvariational algorithms may therefore play an important complementary role to existing variational codes and should be studied further.

Another conclusion, not anticipated by prior analytical work, is that Spitzer's nonvariational algorithm is delicate numerically. The numerical methods require a careful choice of discretization for solving Ampère's law to avoid numerical instability [14]. Even after careful optimization of the discretization, convergence sometimes requires blending of information from earlier iterations and possible filtering of radial Fourier modes to eliminate a slowly growing numerical instability (see Section IV.E below). Such numerical instabilities often arise in strongly nonlinear codes. Similar difficulties have been reported in some variational MHD codes [15] as well as in meteorological codes [16].

The rest of the paper is arranged as follows: In Section II, we give an overview of the PIES code. The following sections expand Section II, giving further details about the physics and about the numerical methods. In Section III, we explain the coordinate systems that are used. In Section IV, we discuss the numerical implementation of the nonvariational method. In Section V, we examine axisymmetric and 3-dimensional MHD equilibria and analyze the convergence properties of the algorithm with respect to various parameters. In Section VI, we give a summary and conclusions. Appendix A discusses how the Fourier coefficients of the magnetic field from the previous iteration can be expressed in the present magnetic coordinates.

Other researchers are working on nonvariational methods for MHD equilibrium problems [17–19], but have not yet published convergence studies of their algorithms. Algorithms similar to Spitzer's have also been studied independently by fluid dynamicists in non-toroidal geometries, solving the Euler equations for inviscid flow in turning channels [20]. However, many of the subtleties of the MHD equations in toroidal domains—large shifts of the magnetic axis, resonances on rational surfaces, and island formation—do not occur in this context.

II. Overview

In this section, we give an overview of the PIES code. Readers primarily interested in the convergence properties of the code should be able to go directly to Section V after reading this section.

Our goal is to calculate solutions of the 3-dimensional MHD equilibrium equations:

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

$$\nabla \times \mathbf{B} = \mathbf{J}, \tag{2}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{3}$$

in a general toroidal domain. Here $\mathbf{J}$ is the current density, $\mathbf{B}$ is the magnetic field, and ∇p is the pressure gradient.

Numerical methods for solving Eqs. (1)–(3) in three dimensions have been traditionally derived variationally using the MHD functional [21, 2]. An alternative approach—that solutions be computed by direct iteration of the equilibrium equations—was originally suggested by Spitzer [9] and by Grad and Rubin [10] although algorithmic details were not given. Interest in this approach was revived by recent calculations of Boozer [11] and of Reiman and Boozer [12]. These authors showed that Spitzer's algorithm had several potential advantages when implemented using magnetic coordinates. In particular, magnetic islands and stochastic fields lines could be treated effectively [12].

The algorithm is simple to state, although subtle to implement [8]. The equilibrium equations, Eqs. (1)–(3), are iterated one after another to find a magnetic field $\mathbf{B}$ and a current density $\mathbf{J}$ that are consistent with given pressure and current profiles and with a given outermost magnetic surface. Various quantities are evaluated along magnetic field lines of $\mathbf{B}$ and then transformed to "near-magnetic" coordinates, which coincide with magnetic coordinates where surfaces exist and which interpolate through regions of islands and stochastic field lines. This allows the code to treat island physics accurately.

The numerical method involves two fundamental steps. The first step is to solve for the current density $\mathbf{J}$ in Eq. (1), given an initial guess for $\mathbf{B}$. This is done by writing the force balance equation as a magnetic differential equation for a stream function ν from which $\mathbf{J}$ can be determined (see Eq. (34) below and Ref. [11]):

$$(\mathbf{B} \cdot \nabla)\nu = p' + \frac{g' + \iota I'}{\mathcal{J}^p}. \tag{4}$$

The variable ρ defines a radial coordinate such that $\rho = 0$ is a magnetic axis of $\mathbf{B}$ while $\rho = 1$ gives the outermost flux surface. The prime $' = d/d\rho$ denotes differentiation with respect to ρ. The functions $p(\rho)$, $g(\rho)$, $I(\rho)$ determine the pressure, the net poloidal current, and the net toroidal current profiles, respectively. The function $\iota(\rho)$ is the rotational transform (winding number) of a magnetic field line defining the magnetic surface labeled by ρ, while $\mathcal{J}^{\rho}$ is the Jacobian between laboratory and near-magnetic coordinates (see Eq. (31) below). This equation is derived in Section IV.B below.

Equation (4) yields both the diamagnetic current density ($\mathbf{J}_{\perp} = \mathbf{B} \times \nabla p/B^2$) and the Pfirsch–Schlüter currents ($\mathbf{J}_{\|} = \mathbf{J} - \mathbf{J}_{\perp}$) upon solution. It can be solved numerically by integrating along field lines of $\mathbf{B}$, by transforming to magnetic coordinates on existing surfaces, and by treating v as constant for stochastic field lines or for lines in magnetic islands [22].

The second step is to solve for the new magnetic field $\mathbf{B}$ that is generated by the current density $\mathbf{J}$ through Ampère's law, Eq. (2). It is straightforward [11] to find a field $\mathbf{h}$ such that $\nabla \times \mathbf{h} = \mathbf{J}$, but $\mathbf{h}$ does not generally satisfy Eq. (3). This suggests trying to find a scalar field u such that

$$\mathbf{b} = \mathbf{h} + \nabla u + \lambda \nabla \phi$$

is divergence free. The new magnetic field $\mathbf{b}$ will then also satisfy Ampère's law since the curl of a gradient is zero. Equation (3) yields a Poisson equations for u:

$$\nabla^2 u = -\nabla \cdot (\mathbf{h} + \lambda \nabla \phi), \tag{5}$$

The constant λ is known in terms of initial conditions, as explained in Section IV.D. Solving for u gives a new magnetic field $\mathbf{b}$ which should be a closer approximation to a solution to the equilibrium equations.

The nonvariational algorithm then consists of repeating these steps many times—calculating new fields $\mathbf{J}$ and $\mathbf{B}$ through the associated scalar fields v and u—until an acceptable level of convergence is attained. The crucial physical advantage of this approach is the formulation in terms of the magnetic differential equation, Eq. (4). This can be accurately solved even for solutions $\mathbf{B}$ with complex field line topology [22].

The numerically subtle part of the nonvariational algorithm lies in solving for u. Overall convergence of the PIES algorithm depends on subtle details of how the Poisson equation is discretized, e.g., whether a conservative discretization is chosen or whether Fourier coefficients are defined on half-integer radial meshes as opposed to integer radial meshes. This numerical sensitivity arises partly because the coordinate system is tied closely to the magnetic field $\mathbf{B}$ and changes with each iteration; partly because the coordinate system is centered on a magnetic axis that also varies from iteration to iteration; and also because some quantities (metric elements, vector components) are singular near the magnetic axis. Further discussion is given in Ref. [14].

For this paper, we will assume that satisfactory numerical methods have been

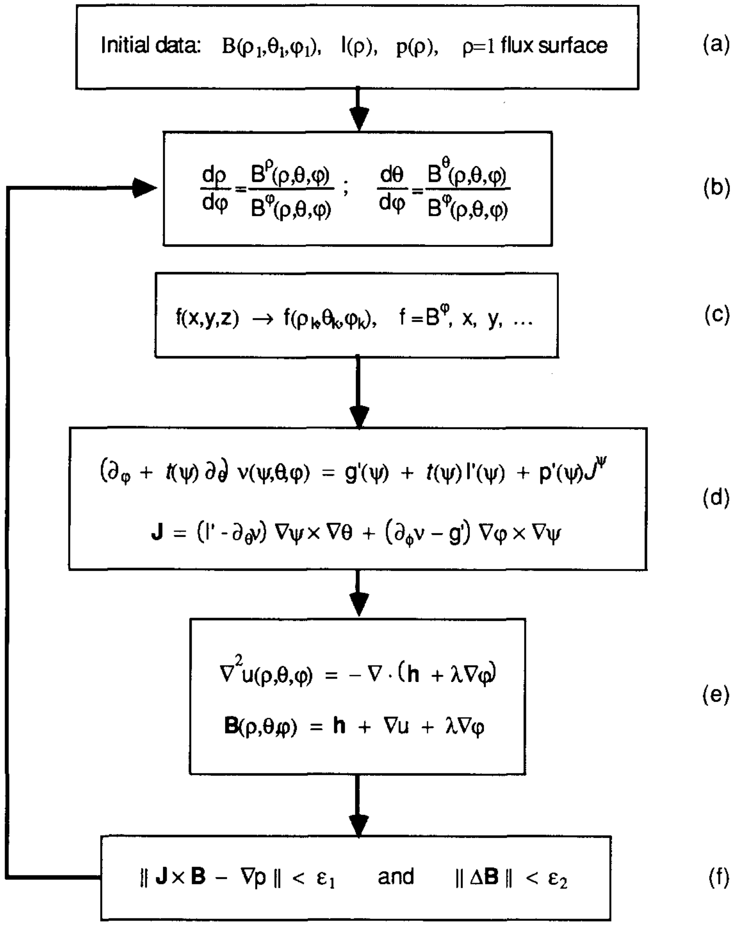
Initial data: $B(\rho_1,\theta_1,\varphi_1)$, $I(\rho)$, $p(\rho)$, $\rho=1$ flux surface
(a)
$\frac{d\rho}{d\varphi}=\frac{B^{\rho}(\rho,\theta,\varphi)}{B^{\varphi}(\rho,\theta,\varphi)}$; $\frac{d\theta}{d\varphi}=\frac{B^{\theta}(\rho,\theta,\varphi)}{B^{\varphi}(\rho,\theta,\varphi)}$
(b)
$f(x,y,z) \rightarrow f(\rho_k,\theta_k,\varphi_k)$, $f = B^{\varphi}$, x, y, ...
(c)
$(\partial_{\varphi} + \iota(\psi)\,\partial_{\theta})\, v(\psi,\theta,\varphi) = g'(\psi) + \iota(\psi)\, I'(\psi) + p'(\psi) J^{\psi}$
$\mathbf{J} = (I' - \partial_{\theta} v)\, \nabla\psi \times \nabla\theta + (\partial_{\phi} v - g')\, \nabla\varphi \times \nabla\psi$
(d)
$\nabla^2 u(\rho,\theta,\varphi) = -\nabla \cdot (\mathbf{h} + \lambda\nabla\varphi)$
$\mathbf{B}(\rho,\theta,\varphi) = \mathbf{h} + \nabla u + \lambda\nabla\varphi$
(e)
$\| \mathbf{J} \times \mathbf{B} - \nabla p \| < \varepsilon_1$ and $\| \Delta\mathbf{B} \| < \varepsilon_2$
(f)

FIG. 1. A schematic flowchart of the central steps in the PIES code: (a) An initial magnetic field is specified in laboratory coordinates (ρ_1, θ_1, ϕ); two surface functions are also given. The outermost flux surface $(\rho = 1)$ is determined by choosing a field line along a symmetry axis through the magnetic axis. (b) Various quantities are evaluated along the field lines of **B** by integrating the field line equations, Eq. (29). The initial points of the field lines lie at uniformly separated values of ρ along a symmetry axis. (c) Various fields such as B^{ϕ}, x, and y are then transformed from laboratory to near-magnetic coordinates (ρ_k, θ_k, ϕ) using the algorithm of Ref. [22], where k is the iteration number. (d) The magnetic differential equation for v is solved and then the current density **J** is determined. (e) A field **h** is calculated such that $\nabla \times \mathbf{h} = \mathbf{J}$. A Poisson equation is then solved for u such that $\mathbf{h} + \nabla u$ is divergence-free, giving a new magnetic field. (f) The residuals of the 3D MHD equations and the correction to the magnetic field are both tested for convergence (ε_1 and ε_2 are small numbers representing desired convergence accuracies). If neither is sufficiently small, the algorithm is iterated again, using the new **B** field for field-line integrations and transformation to near-magnetic coordinates.

derived for solving magnetic differential equations [22] and for solving Poisson equations in toroidal domains [14] so that we can focus on the overall convergence properties of the PIES code, i.e., how these two basic steps interact with each other numerically. We note that these two steps are linear problems and that the nonlinearities of the equilibrium equations arise from integrating along magnetic field lines [22] of **B** and transforming to near-magnetic coordinates. The basic steps of the PIES code are summarized in Fig. 1.

III. Coordinates Systems

In this section, we discuss the coordinate systems used in the PIES code: how fields are represented numerically and how the metric elements are calculated. In principle, the code can use *three* different coordinate systems: a laboratory system (x, y, z) to prescribe initial conditions; "near-magnetic" coordinates (ρ, θ, ϕ) to simplify solution of the magnetic differential equation Eq. (4); and "optimized" coordinates [23] to reduce the effort in solving the Poisson equation (5). In practice, integration along field lines of **B** dominates the computational effort, so only the first two coordinate systems are needed.

Near-magnetic coordinates generalize magnetic coordinates, with constant radial surfaces coinciding with magnetic surfaces where they exist and interpolating through magnetic islands and stochastic regions elsewhere. Upon transforming to near-magnetic coordinates, all fields can be written as a Fourier series in two angular variables, a poloidal angle θ, and a toroidal angle ϕ. The Fourier coefficients are known on a discrete uniform radial mesh labeled by ρ, where $\rho = 0$ is a magnetic axis of **B** and is also the origin of definition for θ, and where $\rho = 1$ is a given outermost flux surface. A Fourier representation for the angles is known to be more efficient and accurate than the use of finite differences [3].

III.A. *Coordinates*

The coordinates (ρ, θ, ϕ) are determined by the field lines of **B** at each iteration of the PIES code [22]. Both ρ and θ change with **B**, while the toroidal angle ϕ remains the same. Fixing ϕ to be the usual cylindrical ϕ removes a gauge ambiguity in θ that exists in magnetic coordinates. It is also a convenient choice for plotting magnetic surfaces in laboratory coordinates.

The toroidal domain in which the MHD equations will be solved is defined in these coordinates by

$$\begin{aligned} &0 \leqslant \rho \leqslant 1; \\ &0 \leqslant \theta \leqslant 2\pi; \\ &0 \leqslant \phi \leqslant 2\pi. \end{aligned} \tag{6}$$

An algorithm for following field lines of a magnetic field **B** and determining (ρ, θ, ϕ)

such that $\rho = 0$ is a magnetic axis and $\rho = 1$ is the given outermost flux surface is given in Ref. [22].

This domain can also be considered to define one period of a device that is not axisymmetric. In that case, we can replace ϕ by $N\phi$ and ∂_ϕ by $N\partial_\phi$ to treat the case of a device with N periods. For simplicity of notation, we shall assume a single period machine, with $N = 1$.

Also for simplicity, we assume that the domain is symmetric under the so-called stellarator symmetry:

$$\theta \rightarrow -\theta; \qquad \phi \rightarrow -\phi. \tag{7}$$

Most stellarator vacuum fields have this symmetry, and the generalization to devices lacking this symmetry is straightforward. With this assumption, scalar fields will have Fourier series that have even or odd parity under this transformation, of the form

$$f(\rho, \theta, \phi) = \sum_{m=0}^{\mathscr{M}} \sum_{n=-\mathscr{N}}^{\mathscr{N}} f_{m,n}(\rho) \cos(n\phi - m\theta), \tag{8}$$

for even parity, and of the form

$$f(\rho, \theta, \phi) = \sum_{m=0}^{\mathscr{M}} \sum_{n=-\mathscr{N}}^{\mathscr{N}} f_{m,n}(\rho) \sin(n\phi - m\theta), \tag{9}$$

for odd parity. Here $\mathscr{M}$ and $\mathscr{N}$ are nonnegative integers that determine the modal resolution in the poloidal and toroidal directions, respectively, and are given as input to the code (see Table I).

TABLE I

Numerical Parameters Needed for the PIES Code

Parameter	Definition
$\mathscr{M}$	Maximum poloidal mode number
$\mathscr{N}$	Maximum toroidal mode number
L	$L+1$ is number of magnetic surfaces
$\bar{m}$	Maximum radial power pulled out of Fourier coefficient
cyl	True for infinite aspect ratio, false otherwise
ftprec	Accuracy of Fourier coeffs from field line following
convg	Convergence precision for MHD equations
rndoff	Effective machine precision ($\|x\| <$ **rndoff** $\Rightarrow x = 0$)
bblnd	Blending parameter for $B^\phi(\rho, \theta, \phi)$
xyblnd	Blending parameter for $x(\rho, \theta, \phi)$, and $y(\rho, \theta, \phi)$
iblnd	Blending parameter for $\iota(\rho)$
s1	Diffusion smoothing element in Eq. (56)
s2	Antidiffusion smoothing element in Eq. (56)

Note. These are the most important of the many parameters set in the code. See Refs. [22, 14] for more details and a discussion of other parameters.

The Fourier coefficients $f_{m,n}(\rho)$ are assumed to be known on either of two uniform radial meshes, an integer mesh commensurate with $\rho = 0$ and with $\rho = 1$,

$$\rho_l = l d\rho, \qquad l = 0, ..., L; \tag{10}$$

or a half-integer mesh that is shifted half a radial mesh spacing with respect to the integer mesh,

$$\rho_{l+1/2} = (l + 1/2)\, d\rho, \qquad l = 0, ..., L - 1. \tag{11}$$

The radial mesh spacing is $d\rho = 1/L$, where L is some positive integer determining the radial resolution. We will use a superscript l to denote the value of a Fourier coefficient at the lth mesh point: $a^l_{m,n} = a_{m,n}(l d\rho)$, with a similar notation for the half-integer mesh.

Each scalar field therefore has $(L+1)(\mathcal{M}+1)(2\mathcal{N}+1)$ degrees of freedom and is stored as a Fortran 77 real array of the form **f(O:M, −N:N, O:L)**. In the absence of the stellarator symmetry, m would vary from $-\mathcal{M}$ to $\mathcal{M}$ and cosine and sine modes would both be present; four times as many degrees of freedom would be required. We note that the representations, Eqs. (8) and (9), have redundant coefficients since the terms

$$f_{0,n}(\rho) = \pm f_{0,-n}(\rho), \tag{12}$$

are identical for $n > 0$. This redundancy is intentional and serves two useful purposes. First, it allows the parity of each array to be tested, which is useful for verifying the correctness of a long calculation that produced that array. Second, it allows vectorization of Fortran **do** loops using the natural bounds, **do m = O, M**; **do n = −N, N**; and **do *l* = O, L**.

The near-magnetic coordinates are related at each iteration to a laboratory coordinate system. Instead of using Cartesian coordinates (X, Y, Z), it is more useful computationally to use a rotating laboratory coordinate system (x, y, z), which is shown in Fig. 2. These coordinates require fewer Fourier coefficients when

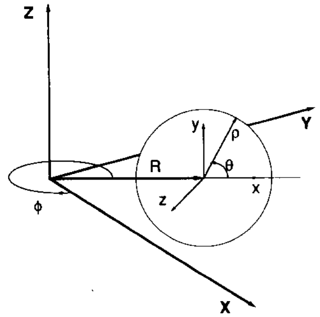

FIG. 2. Schematic relation between the Cartesian coordinates (X, Y, Z), the laboratory coordinates (x, y, z), and the magnetic coordinates (ρ, θ, ϕ). (x, y, z) forms a right-handed orthogonal coordinate system, while (ρ, θ, ϕ) is right-handed but generally not orthogonal or separable. R is the major radius.

expressed in magnetic coordinates [6]. For finite aspect ratio domains, $z = R\phi$, where R is the major radius and ϕ is the toroidal angle of a cylindrical coordinate system. For finite aspect ratio domains, z becomes the usual constant Cartesian coordinate Z. An arbitrary vector can then be written in the form

$$\mathbf{X}(\rho, \theta, \phi) = (R + x(\rho, \theta, \phi)\, \hat{\mathbf{x}}) + y(\rho, \theta, \phi)\, \hat{\mathbf{y}}, \tag{13}$$

where the unit vectors are given in Cartesian coordinates by

$$\hat{\mathbf{x}} = \cos(\phi)\, \hat{X} - \sin(\phi)\, \hat{Y}, \tag{14}$$

$$\hat{\mathbf{y}} = \hat{Z}, \tag{15}$$

$$\hat{\mathbf{z}} = \partial_\phi \hat{\mathbf{x}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} = -\sin(\phi)\, \hat{X} - \cos(\phi)\, \hat{Y}. \tag{16}$$

These definitions are appropriate for a finite aspect ratio torus. For infinite aspect ratio cylinders, the coordinates are Cartesian,

$$\mathbf{X}(\rho, \theta, \phi) = x(\rho, \theta, \phi)\hat{\mathbf{x}} + y(\rho, \theta, \phi)\, \hat{\mathbf{y}} + R\phi\hat{\mathbf{z}} \tag{17}$$

$$\hat{\mathbf{x}} = \hat{Y}; \qquad \hat{\mathbf{y}} = \hat{Z}; \qquad \hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} = \hat{X}, \tag{18}$$

where R is now the effective major radius of the cylinder ($2\pi R$ is the length of the magnetic axis). An input parameter **cyl** is needed to tell the code whether the domain has finite **(cyl = NO)** or infinite aspect ratio **(cyl = YES)**.

At the beginning of each iteration of the PIES code, the coordinates x and y are expressed in Fourier series of the forms given by Eqs. (8) and (9), respectively. This is achieved [22] by evaluating x and y along fields lines of the magnetic field **B**.

Once these Fourier series are known, we can calculate the metric elements of the near-magnetic coordinate system defined implicitly by **B**. This requires evaluating numerical derivatives of the coordinate vector $\mathbf{x}(\rho, \theta, \phi)$ and convoluting derivatives in pairs to avoid aliasing. The numerical derivatives require special care to obtain accuracy and numerical stability near the magnetic axis, as discussed in the next section.

III.B. *Derivatives of Fourier Coefficients*

Numerical approximations to derivatives first arise in evaluating the Fourier coefficients of the covariant coordinate vectors

$$\mathbf{e}_\rho = \mathbf{X}_{,\rho} = x_{,\rho}\hat{\mathbf{x}} + y_{,\rho}\hat{\mathbf{y}} \tag{19}$$

$$\mathbf{e}_\theta = \mathbf{X}_{,\theta} = x_{,\theta}\hat{\mathbf{x}} + y_{,\theta}\hat{\mathbf{y}} \tag{20}$$

$$\mathbf{e}_\phi = \mathbf{X}_{,\phi} = \begin{cases} x_{,\phi}\hat{\mathbf{x}} + y_{,\phi}\hat{\mathbf{y}} + (R + x)\hat{\mathbf{z}} & \text{finite aspect ratio} \\ x_{,\phi}\hat{\mathbf{x}} + y_{,\phi}\hat{\mathbf{y}} + R\hat{\mathbf{z}} & \text{infinite aspect ratio,} \end{cases} \tag{21}$$

which are needed in constructing the covariant metric elements $g_{ij} = \mathbf{e}_i \cdot \mathbf{e}_j$. Here, a comma notation is being used to indicate partial derivatives with respect to a variable, e.g., $\mathbf{X}_{,\rho} = \partial\mathbf{X}/\partial\rho$, with similar expressions for θ and ϕ derivatives.

The angular derivatives are obtained in the usual way, multiplying Fourier coefficients by mode numbers. The radial derivatives of Fourier coefficients require more careful consideration. Straightforward finite difference approximations to the radial derivatives of Fourier coefficients $f_{m,n}(\rho)$ give large relative errors near the magnetic axis for large poloidal mode numbers m, of $O(1)$ instead of $O(d\rho^2)$. We found it necessary to modify the finite difference expressions to avoid this problem.

The reason for the large relative error in the radial derivative is that second-order accurate finite differences are reasonable only when $f_{m,n}(\rho)$ is locally well approximated by a quadratic curve. This is generally not the case in polar-like coordinates near the polar origin [14]. Fourier coefficients vary near the magnetic axis as a power of ρ times a power series in ρ^2,

$$f_{m,n}(\rho) \simeq \rho^{e_f(m)}(f^0_{m,n} + f^2_{m,n}\rho^2 + \cdots), \qquad \rho \ll 1, \tag{22}$$

where $f^0_{m,n}, f^2_{m,n}, \ldots$, are constants and where $e_f(m)$ is an integer function that depends on the scalar f. For analytic functions $\mathscr{A}$, $e_{\mathscr{A}}(m) = m$. (Analytic functions $\mathscr{A}(\rho, \theta, \phi) = \mathscr{A}(\xi, \eta, \phi)$ have bounded Taylor series near the magnetic axis when expressed in the Cartesian-like variables $\xi = \rho\cos(\theta)$ and $\eta = \rho\sin(\theta)$.) For metric elements and other, more complicated, combinations of analytic functions, $e_f(m)$ is approximately m for large m.

Equation (22) suggests that the large relative error in radial derivatives near $\rho = 0$ can be reduced by introducing radial factors $R(f)_m(\rho)$ whose derivative can be taken analytically, and such that $f_{m,n}(\rho)/R(f)_m(\rho)$ is more locally quadratic. One reasonable choice is

$$R(f)_m(\rho) = \rho^{\min(e_f(m), \bar{m})}. \tag{23}$$

The radial factor depends on both the mode number and the particular function f. The integer $\bar{m}$, which satisfies $1 \leqslant \bar{m} \leqslant m$, prevents $R(f)_m(\rho)$ from becoming too small for large m and small ρ. This cutoff is necessary when solving Ampère's law, whose discretized form can become ill-conditioned near the magnetic axis if large radial powers are allowed. In practice, $\bar{m} = 2$ gives satisfactory results, balancing the need to obtain increased accuracy near the magnetic axis with the need to avoid ill-conditioning with tiny numbers.

We then define the radial derivative at the mesh value $\rho_l = l\,d\rho$ to be

$$\left(\frac{\partial}{\partial\rho} f\right)^l_{m,n} = \left(\frac{\partial}{\partial\rho} R(f)_m\right)^l \frac{f^l_{m,n}}{R(f)^l_m} + R(f)^l_m\,\delta_\rho\left(\frac{f_{m,n}}{R(f)_m}\right)^l. \tag{24}$$

Here δ_ρ represents the appropriate second-order accurate finite-difference operator. On the integer mesh, it takes the form

$$(\delta_\rho a_{m,n})^l = \begin{cases} 2(a^2_{m,n} - a^1_{m,n})/(3d\rho), & l = 1, \\ (a^{l+1}_{m,n} - a^{l-1}_{m,n})/(2d\rho), & 1 < l < L, \\ (3a^l_{m,n} - 4a^{l-1}_{m,n} + a^{l-2}_{m,n})/(2d\rho), & l = L, \end{cases} \tag{25}$$

with a similar expression for the derivative on the half-integer mesh in terms of the values on the integer mesh.

The second-order accurate two-point discretization next to the magnetic axis ($l=1$) is motivated by Eq. (22), which shows that the normalized Fourier coefficient $f_{m,n}/R(f)_m$ has no term linear in ρ for $\rho \ll 1$. This is correct when $e_f(m) - \min(\bar{m}, e_f(m))$ is even, so that the expansion Eq. (22) is valid. When this difference is odd, a discretization based on an expansion of the form $f^1_{m,n}\rho + f^3_{m,n}\rho^3 + \cdots$ is required. A one-sided derivative at $\rho = 1$ is used since we assume that values of fields are not known exterior to the domain D.

IV. The Nonvariational Method

In this section, details are given of the PIES algorithm: how equilibrium solutions to the MHD equations can be obtained by iterating in carefully chosen coordinates. The description is more complete than that given in our earlier paper [8], at a level more appropriate for obtaining a working code.

We first discuss how the force balance equation, Eq. (1), can be reduced to the magnetic differential equation, Eq. (4), for the stream function for the current density [24, 11]. Then we discuss how Ampère's law can be solved to obtain a field **h** whose curl gives this current [11]. This field is generally not solenoidal but can be made so by adding the gradient of a scalar field u (see Eq. (5)). We next discuss how to obtain u, which requires the solution of a Poisson equation with Neumann boundary conditions in a nonseparable coordinate system [14]. Finally, we disucuss blending and filtering of Fourier coefficients, which are sometimes needed to obtain convergence.

IV.A. *Boundary Conditions and Surface Functions*

We want to solve the MHD equations, Eqs. (1)–(3), in the domain defined by Eq. (6). The MHD equations have to be supplemented with boundary conditions and by two surface functions (or profiles) [21]. A fixed perfectly conducting boundary is assumed so that the outermost magnetic surface is known:

$$\mathbf{B} \cdot \nabla p = B^\rho = 0, \qquad \text{for} \quad \rho = 1. \tag{26}$$

This boundary condition enters the Poisson equation related to Ampère's law, and implies a Neumann boudary condition (see Secton IV.D).

A second boundary condition is also required in solving Ampère's law to pin down the multi-valuedness of the scalar solution to the Poisson equation (this is needed since the domain D is not simply connected and a Neumann boundary condition does not specify a unique solution). There are two natural boundary conditions that can be imposed: conservation of net toroidal flux

$$\int \mathbf{B} \cdot d\mathbf{S}_\phi = \int_{\phi=\phi_0} \mathbf{B} \cdot (\mathscr{J}^\rho \nabla\phi)\, d\rho\, d\theta = \text{constant}, \tag{27}$$

where the integration goes over a surface of constant toroidal angle $\phi = \phi_0$ that intersects the domain; or conservation of net external poloidal current (i.e., current through the external coils that thread the torus):

$$\int \mathbf{J} \cdot d\mathbf{S}_\theta = \int \mathbf{B} \cdot dl_\phi = \int_{\rho = 1,\, \theta = \theta_0} \mathbf{B} \cdot (\mathcal{J}^{\rho} \nabla\rho \times \nabla\theta)\, d\phi = \text{constant}, \tag{28}$$

where the integration now goes over the boundary of the interior hole of the domain. Equation (27) involves an integration over ρ, which couples all the radial surfaces together. Equation (28) couples only the outermost three radial surfaces and leads to a nearly block tridiagonal matrix for the Poisson equation. This is more convenient numerically. It is also physically more similar to the conditions of many experiments.

Besides boundary conditions, two surface functions must be specified to obtain a unique solution to the MHD equations [21]. It is convenient to specify the pressure profile $p(\rho)$ and a current profile, either the net toroidal current $I(\rho)$, or the net poloidal current $g(\rho)$.

The PIES code therefore begins with an initial magnetic field **B**, a pressure profile $p(\rho)$, and a current profile ($I(\rho)$ or $g(\rho)$); see Fig. 1. The contravariant components of the magnetic field are first given in the usual pseudo-toroidal laboratory coordinate system. The fields x, y, and B^ϕ are evaluated along $L+1$ field lines as functions of ϕ by integrating the equations:

$$\frac{d\rho}{d\phi} = \frac{B^\rho(\rho,\theta,\phi)}{B^\phi(\rho,\theta,\phi)}; \qquad \frac{d\theta}{d\phi} = \frac{B^\theta(\rho,\theta,\phi)}{B^\phi(\rho,\theta,\phi)}, \tag{29}$$

where we make the important assumption that B^ϕ is nonzero in the toroidal domain. The $L+1$ starting points for the field lines define a uniformly spaced radial mesh along symmetry axis connecting the magnetic axis $\rho = 0$ and the boundary $\rho = 1$. This symmetry axis exists by our assumption of the stellarator symmetry, Eq. (7).

Applying the algorithm of Ref. [22], the fields x, y, and B^ϕ become known Fourier expansions in near-magnetic coordinates (ρ, θ, ϕ). The algorithm also yields the rotational transform $\iota(\rho)$, which is the winding number of the field line in the magnetic surface defined by that field line. The Fourier coefficients of all these quantities are determined by the line-following algorithm to an absolute accuracy **ftprec** which is an input parameter, typically of order 10^{-6}. The total external poloidal current, Eq. (28), is held constant during each iteration. The numerical data needed by the code are summarized in Table I, while the physics data are summarized in Table II.

Starting with these expansions in near-magnetic coordinates, we can then proceed to the iteration of the MHD equations, Eq. (1)–(3). The equations are more simply stated by using another radial variable, ψ, which arises naturally from the canonical form of a magnetic field that has magnetic surfaces [21]:

$$\mathbf{B} = \nabla\psi \times \nabla\theta + \iota\, \nabla\phi \times \nabla\psi. \tag{30}$$

TABLE II

Physics Data Needed as Input to the PIES Code

Data	Definition
R	Major radius of domain
N	Number of periods in stellarator
$B^x(x, y, z)$	x component of initial **B** field in lab coordinates
$B^y(x, y, z)$	y component of initial **B** field in lab coordinates
$B^z(x, y, z)$	z component of initial **B** field in lab coordinates
$dp/d\rho$	Derivative of pressure profile
$dI/d\rho$	Derivative of net toroidal current profile
$dg/d\rho$	Derivative of net poloidal current profile

Note. Either I' or g' is specified, with the other profile being determined by Eq. (36). The magnetic axis and outermost magnetic surface are determined by field line integrations, as described in Ref. [22]. The locations of the axis and boundary are the only information retained from **B**.

The variable $\psi(\rho)$ is a nonnegative montonic function of the radial variable ρ and is easily seen to be $1/2\pi$ times the toroidal flux enclosed by a surface labeled by ρ. The rotational transform, $\iota(\rho)$, is the surface quantity mentioned previously, giving the average rate of poloidal rotation of the field line.

The radial variable ψ is useful provided the relation $\psi(\rho)$ is known. This function can be determined on existing magnetic surfaces by averaging the ratio of the Jacobians between Cartesian and near-magnetic coordinates, with ρ and ψ as independent radial variables respectively. The ρ-Jacobian, is known in terms of derivatives and convolutions of the cylindrical coordinates x and y:

$$\begin{aligned}\mathscr{J}^{\rho} &= \frac{\partial(X, Y, Z)}{\partial(\rho, \theta, \phi)} = \mathbf{e}_{\rho} \times \mathbf{e}_{\theta} \cdot \mathbf{e}_{\phi} \\ &= \begin{cases} (x+R)(x_{,\rho} y_{,\theta} - x_{,\theta} y_{,\rho}) & \text{finite aspect ratio,} \\ R(x_{,\rho} y_{,\theta} - x_{,\theta} y_{,\rho}) & \text{infinite aspect ratio.} \end{cases}\end{aligned} \tag{31}$$

From Eq. (30) we see that the Jacobian between near-magnetic coordinates with ψ as the independent variable is determined by the B^{ϕ} component of the magnetic field:

$$\begin{aligned}\mathscr{J}^{\psi} &= \frac{\partial(X, Y, Z)}{\partial(\psi, \theta, \phi)} = \frac{1}{\nabla\psi \times \nabla\theta \cdot \nabla\phi} = \frac{1}{\mathbf{B} \cdot \nabla\phi} = \frac{1}{B^{\phi}} \\ &= \sum_{m,n} \mathscr{J}^{\psi}_{m,n}(\psi) \cos(n\phi - m\theta).\end{aligned} \tag{32}$$

The Fourier coefficients of $1/B^{\phi}(\psi, \theta, \phi)$ are known by having evaluated B^{ϕ} along field lines of **B**, using the transformation to near-magnetic coordinates of Ref. [22], and solving a set of linear equations for the Fourier coefficients of the inverse Fourier series. The averaged ratio of the two Jacobians, then gives

$$\frac{d\psi}{d\rho} = \langle \mathscr{J}^{\rho}(1/\mathscr{J}^{\psi}) \rangle = \langle \mathscr{J}^{\rho} B^{\phi} \rangle, \tag{33}$$

where the brackets $\langle \cdots \rangle$ denote an average over poloidal and toroidal angles. Thus $\psi'(\rho)$ is the $m = n = 0$ mode of $\mathscr{J}^{\rho} B^{\phi}$.

IV.B. *Calculation of the Current Density* **J**

From $\nabla \times \mathbf{B} = \mathbf{J}$, we deduce that the current density **J** must be solenoidal. From $\mathbf{J} \times \mathbf{B} = \nabla p$, we see that **J** must lie on magnetic surfaces [21]. The current density can therefore also be represented in a canonical form similar to Eq. (30),

$$\mathbf{J} = \left(I'(\psi) - \frac{\partial v}{\partial \theta} \right) \nabla\psi \times \nabla\theta + \left(\frac{\partial v}{\partial \phi} - g'(\psi) \right) \nabla\phi \times \nabla\psi, \tag{34}$$

where we are now using ψ as the radial variable. The single-valued scalar $v(\psi, \theta, \phi)$ is a degree of freedom that is allowed under the constraints $\nabla \cdot \mathbf{J} = 0$ and $\mathbf{J} \cdot \nabla\psi = 0$. By integrating $\mathbf{J} \cdot \nabla\phi$ over the volume enclosed by a magnetic surface labeled by ψ, we see that $2\pi I(\psi)$ is the net toroidal current profile. Similarly, integrating the quantity $\mathbf{J} \cdot \nabla\theta$ over a surface labeled by θ shows that $2\pi g(\psi)$ is the net poloidal current profile external to the magnetic surface.

A magnetic differential equation for $v(\psi, \theta, \phi)$ is obtained [24] by substituting Eqs. (30) and (34) into the force balance equation, Eq. (1),

$$(\mathbf{B} \cdot \nabla) v = \frac{g' + tI'}{\mathscr{J}^{\psi}} + \frac{dp}{d\psi},$$

or

$$\left(\frac{\partial}{\partial \phi} + t \frac{\partial}{\partial \theta} \right) v = g' + tI' + \frac{dp}{d\psi} \mathscr{J}^{\psi}, \tag{35}$$

where the $'$ denotes differentiation with respect to ψ.

By integrating both sides of this equation over θ and ϕ and using the fact that v is single-valued and hence periodic, we see that Eq. (35) has a solution only if

$$g' + tI' + \frac{dp}{d\psi} \mathscr{J}^{\psi}_{0,0} = 0, \tag{36}$$

where $\mathscr{J}^{\psi}_{0,0}$ is the $m = n = 0$ mode of the Fourier representation of $\mathscr{J}^{\psi}$ (see Eq. (32)). This relation allows us to specify either I' or g' and to solve for the other current profile. This relation between the two profiles was originally given by Kruskal and Kulsrud [21] in the form $dp\, dV = d\chi\, dI - d\psi\, dg$, where V is the volume enclosed by a magnetic surface labeled by ψ, and where $\chi = t\psi$ is the poloidal flux enclosed by the same surface.

The solution to the equation for v, Eq. (35), in terms of the Fourier coefficients of $\mathcal{J}^{\psi}$ (Eq. (32)), is simply

$$v(\psi, \theta, \phi) = \frac{dp}{d\psi} \sum_{m,n}{}' \frac{\mathcal{J}^{\psi}_{m,n}}{(n - \iota m)} \sin(n\phi - m\theta), \tag{37}$$

where the $'$ indicates that the $m = n = 0$ term is omitted. Substituting this into the expression for **J**, Eq. (34), then gives the canonical expression for the current density:

$$\begin{aligned} \mathbf{J} = \nabla\psi \times \nabla\theta \left(I'(\psi) + \frac{dp}{d\psi} \sum_{m,n}{}' \frac{m\mathcal{J}^{\psi}_{m,n}}{(n - \iota m)} \cos(n\phi - m\theta) \right) \\ + \nabla\phi \times \nabla\psi \left(-g'(\psi) + \frac{dp}{d\psi} \sum_{m,n}{}' \frac{n\mathcal{J}^{\psi}_{m,n}}{(n - \iota m)} \cos(n\phi - m\theta) \right). \end{aligned} \tag{38}$$

Together with Eq. (36), this finishes the first step of the PIES algorithm, calculating the total current density **J** in terms of **B**, the pressure profile p', a current profile, and a known outermost flux surface.

IV.C. *Solution of Ampère's Law*

We next discuss how to solve Ampère's law, Eq. (2), for a new solenoidal magnetic field **b**. As observed by Boozer [11], it is simple to find a field **h** such that $\nabla \times \mathbf{h} = \mathbf{J}$. In fact, if **h** is not solenoidal, there is an infinity of such fields that all differ by the gradient of some scalar. Ampère's law in covariant form leads to the following three equations on each radial surface:

$$n(h_\theta)_{m,n} + m(h_\phi)_{m,n} = (\mathcal{J}^\rho J^\rho)_{m,n}, \tag{39}$$

$$n(h_\rho)_{m,n} - \partial_\rho(h_\phi)_{m,n} = (\mathcal{J}^\rho J^\theta)_{m,n}, \tag{40}$$

$$m(h_\rho)_{m,n} + \partial_\rho(h_\theta)_{m,n} = (\mathcal{J}^\rho J^\varphi)_{m,n}. \tag{41}$$

This suggest two natural gauges. One is to set h_θ or h_ϕ to zero for $m > 0$. This gives the following solution (which generalizes Boozer's gauge [11] to the case of finite $\mathcal{J}^\rho$):

$$\langle (h_\rho)_{m,n}, (h_\theta)_{m,n}, (h_\phi)_{m,n} \rangle = \begin{cases} \dfrac{1}{m} \langle \mathcal{J}^\rho J^\phi_{m,n}, 0, \mathcal{J}^\rho J^\rho_{m,n} \rangle & \text{for} \quad m > 0; \\ \dfrac{1}{n} \langle \mathcal{J}^\rho J^\theta_{m,n}, \mathcal{J}^\rho J^\rho_{m,n}, 0 \rangle & \text{for} \quad m = 0 \text{ and } n \neq 0; \\ \left\langle 0, \displaystyle\int_0^\rho \mathcal{J}^\rho J^\phi_{0,0}\, d\rho, -\int_0^\rho \mathcal{J}^\rho J^\theta_{0,0}\, d\rho \right\rangle & \text{for} \quad m = n = 0. \end{cases} \tag{42}$$

For fields with magnetic surfaces (Eq. (38)), we can write this out explicitly:

$$\mathbf{h} = \left(\frac{dp}{d\rho} \sum_{m,n}' \frac{\mathcal{J}^{\psi}_{m,n}}{(n - \iota m)} \sin(n\phi - m\theta)\right) \nabla\rho$$
$$+ (I(\rho) - I(0))\, \nabla\theta + (g(\rho) - g(0))\, \nabla\phi. \tag{43}$$

Another choice of gauge is $(h_\rho)_{m,n} = 0$ for all m and n, which gives the following solution to Ampère's law:

$$\langle (h_\rho)_{m,n}, (h_\theta)_{m,n}, (h_\phi)_{m,n} \rangle$$
$$= \left\langle 0, \int_0^\rho \mathcal{J}^\rho J^\phi_{m,n}\, d\rho, -\int_0^\rho \mathcal{J}^\rho J^\theta_{m,n}\, d\rho \right\rangle. \tag{44}$$

For fields with magnetic surfaces, we can again write this out explicitly:

$$\mathbf{h} = \left[I(\rho) - I(0) + \sum_{m,n}' m \int_0^\rho \left(\frac{dp}{d\rho} \frac{\mathcal{J}^{\psi}_{m,n}}{n - \iota m}\right) d\rho \cos(n\phi - m\theta) \right] \nabla\theta$$
$$+ \left[g(\rho) - g(0) - \sum_{m,n}' n \int_0^\rho \left(\frac{dp}{d\rho} \frac{\mathcal{J}^{\psi}_{m,n}}{n - \iota m}\right) d\rho \cos(n\phi - m\theta) \right] \nabla\phi. \tag{45}$$

For circular magnetic surfaces, this gives $h^\rho = 0$, so that this gauge is appropriate for equilibria with nearly circular magnetic surfaces.

Although both gauges are analytically equivalent in that both satisfy $\nabla \times \mathbf{h} = \mathbf{J}$, they are different numerically. In particular, Eq. (43) involves only the radial integrals of $m = n = 0$ modes, while Eq. (45) involves radial integrals of higher mode numbers.

Both choices of gauge for $\mathbf{h}$ require numerical approximations to integrals of Fourier coefficients. Two different numerical approximations were tried. The first method was the usual trapezoidal rule. The second method evaluated integrals by analytically integrating the interpolating splines of the integrands, using "not-a-knot" boundary conditions [25] at $\rho = 0$ and at $\rho = 1$. Numerical experiments showed that the convergence of the PIES code was only weakly dependent on the choice of integration algorithm. In particular, the trapezoidal rule gave satisfactory results. This is an important observation for applications of the code to equilibria with islands, since global integration methods using splines or radial polynomials are awkward to use in the presence of islands, for which Fourier coefficients can vary rapidly over short radial intervals.

A potential difficulty with solving directly for $\mathbf{h}$ in Ampère's law is that this field remains of $O(1)$ as the iterations proceed. This could lead to numerical convergence problems since, as we will see in the next section, the ρ component of the new magnetic field has the form

$$b^\rho = \nabla\rho \cdot (\mathbf{h} + \nabla u + \lambda\, \nabla\phi),$$

and this must vanish on all magnetic surfaces. If **h** remains of $O(1)$, then the scalar field u also remains of $O(1)$ by Eq. (5), and a cancellation of large numbers must take place to allow b^{ρ} to vanish.

Because this cancellation can possibly magnify numerical errors and slow convergence, we explored a modification of the PIES algorithm, in which the **B** field at the beginning of the iteration was subtracted from the right side of Ampère's law,

$$\nabla \times \mathbf{h} = \mathbf{J} - \nabla \times \mathbf{B}, \tag{46}$$

so that **h** becomes the *correction* to the magnetic field from the previous iteration. The field **h** and solution u now go to zero as the code converges.

Appendix A discusses how the magnetic field **B** can be expressed in terms of its own magnetic coordinates, so that $\nabla \times \mathbf{B}$ in Eq. (46) is a meaningful expression. Equations (42) and (44) can then be modified to subtract off this new field. The consequence of subtracting $\nabla \times \mathbf{B}$ is discussed in Section V.

IV.D. *Solving a Poisson Equation for u*

The field **h** of the previous section is typically not divergence-free, and so does not satisfy the last of the MHD equations, Eq. (3). It can be made solenoidal—without changing its curl—by adding the gradient of some scalar. A magnetic field **b** whose curl is **J** is then given by

$$\mathbf{b} = \mathbf{h} + \nabla u + \lambda \nabla \phi, \tag{47}$$

where u is a single-valued scalar field to be determined and where the constant λ fixes the possible multivalued correction to u. It is determined by the conservation of the total external poloidal current, Eq. (28):

$$\int_{\rho = 1, \theta = \theta_0} \mathbf{B}^0 \cdot dl = \int_{\rho = 1, \theta = \theta_0} (\mathbf{h} + \nabla u + \lambda \nabla \phi) \cdot dl.$$

The integrals go over a closed loop on the outermost magnetic surface and are independent of θ_0 provided there is no net poloidal current on the outermost magnetic surface. Since u is single-valued, the line integral over a closed loop of its gradient is zero. The left side is known from the initial field $\mathbf{B}^0$ and **h** is known from Eq. (43) or from Eq. (45). We then find

$$\lambda = (B^0_{\phi})^l_{0,0} - (h_{\phi})^l_{0,0}, \tag{48}$$

which involves the $m = n = 0$ Fourier components of B^0_{ϕ} and of h_{ϕ} at $\rho = 1$ $(l = L)$.

The requirement that Eq. (47) satisfies Eq. (3) leads to a Poisson equation for u:

$$\nabla^2 u = -\nabla \cdot (\mathbf{h} + \lambda \nabla \phi). \tag{49}$$

The boundary condition is determined by our assumption of a fixed perfectly conducting boundary, $b^\rho = 0$ for $\rho = 1$, Eq. (26). This leads to a Neumann condition on u:

$$\nabla\rho \cdot \nabla u = -(\mathbf{h} + \lambda\, \nabla\phi) \cdot \nabla\rho \qquad \text{for} \quad \rho = 1. \tag{50}$$

Equations (49) and (50) have to be solved numerically for the Fourier coefficients of u on each magnetic surface. A solution always exists since the usual integral condition for the existence of a solution u of a Poisson equation with a Neumann boundary condition [26] is trivially satisfied: the volume integral of the right side of Eq. (49) is equal to the surface integral of the right side of Eq. (50).

The term $\nabla\phi$ on the right side of Eq. (49) is retained for the following reason. Although ϕ is harmonic, so that $\nabla \cdot \nabla\phi = \nabla^2\phi = 0$, the discrete operator approximating the Laplacian does not generally kill ϕ. We found that retaining the $\nabla\phi$ term on the right side increased the numerical stability of the PIES code. The reason is that a similar gradient term appears in Eq. (50), so that the right side of the discrete equations in the interior of the domain is more consistent with the right side of the boundary condition.

The details of how Eqs. (49) and (50) are discretized and solved are given in Ref. [14]. As is pointed out in that paper, the main subtlety is choosing a dis[illegible] instabilities near the outermost magnetic surface can be eliminated. The traditionally difficult region near the magnetic axis can be controlled by using leading radial factors when performing derivatives or averages, along the lines discussed in Section III.B. A weak slowly growing radial instability sometimes remains but can be eliminated by blending or filtering of Fourier coefficients, as discussed in the next section.

IV.E. *Blending and Filtering of Fields*

This completes one cycle of iterating through the MHD equations. In general, many cycles are needed, the exact number depending on the choice of initial conditions, the numerical resolution, the desired accuracy, and the values of physical parameters. Since the PIES algorithm is a Picard iteration scheme, in which the new fields are fed back as input to the next cycle of iteration, convergence can be slow or even unstable without some massaging of the fields before entering the next iteration. We have found it sometimes useful to blend the Fourier coefficients at the end of one iteration with those of the previous iteration (which aids convergence by preventing oscillation or overshooting). It is also sometimes useful to filter certain Fourier coefficients radially. This eliminates a weak instability of high frequency noise that can appear after many iterations.

Since the coordinates and the representations of fields are changing at each iteration with **B**, it is important to choose fields for blending that are invariant under coordinate transformations. One good set of fields for blending is

$$B^\phi(\rho, \theta, \phi), \qquad x(\rho, \theta, \phi), \qquad y(\rho, \theta, \phi), \qquad \text{and} \qquad t(\rho). \tag{51}$$

This is roughly equivalent to blending the magnetic field since **B** is known in terms of these quantities,

$$\mathbf{B} = B^{\phi}\left(\frac{\partial \mathbf{X}}{\partial \phi} + t\frac{\partial \mathbf{X}}{\partial \theta}\right).$$

This is the expression for **B** in covariant basis vectors (Eq. (30)), using the fact that $1/\mathcal{J}^{\psi} = B^{\phi}$ by Eq. (32), and using Eqs. (13)–(18) to express **X** in terms of x and y.

We introduce three different blending coefficients for these fields: **bblnd**, **xyblnd**, and **iblnd**, and blend the Fourier coefficients at the latest iteration $(K+1)$ with those of the previous one (K), using a simple relaxation scheme:

$$B^{\phi}(K+1) \leftarrow B^{\phi}(K+1) + \mathbf{bblnd}(B^{\phi}(K) - B^{\phi}(K+1)), \tag{52}$$

$$\mathbf{x}(K+1) \leftarrow \mathbf{x}(K+1) + \mathbf{xyblnd}(\mathbf{x}(K) - \mathbf{x}(K+1)), \tag{53}$$

$$t(K+1) \leftarrow t(K+1) + \mathbf{iblnd}(t(K) - t(K+1)). \tag{54}$$

It is understood that the blending is applied to each separate Fourier coefficient. More sophisticated blending algorithms could also be used that give faster convergence by using information from earlier iterations [27].

In addition to blending, we also implemented a subroutine for radial filtering of the Fourier coefficients of certain fields such as **X** and **B**. A simple method for removing this high frequency noise was given by Shapiro [28]. The idea is to use a radial diffusion operator to eliminate the highest frequency noise, and then apply an "antidiffusion" operator to restore the amplitudes of the lower frequency radial modes. If we define the operator

$$F_s[f^{l}_{m,n}] = (1-s)\, f^{l}_{m,n} + \frac{s}{2}\,(f^{l+1}_{m,n} + f^{l-1}_{m,n}), \tag{55}$$

then F_s is diffusive for real numbers $s > 0$ and antidiffusive for $s < 0$. Following Shapiro, we use a two-piece filter on each Fourier coefficient:

$$\bar{f}^{l}_{m,n} = F_{s2}[F_{s1}[f^{l}_{m,n}]]. \tag{56}$$

It was shown by Shapiro that the values $s1 = \frac{1}{2}$ and $s2 = -\frac{1}{2}$ are optimal choices for these parameters in that the highest frequency mode is removed with minimal amplitude or phase change of smaller frequency radial modes, and that no other linear filter acting on five neighboring points can do as well [28]. Our experience is that weak damping usually suffices, with the antidiffusing only slightly improving convergence.

Equation (56) has to be modified near $\rho = 0$ and $\rho = 1$ so that the filtering is consistent with known boundary conditions on the Fourier coefficients. These boundary conditions are that the correct analytic behavior be recovered by the filtered coefficients and that information on the outermost magnetic surface is assumed to be more accurate than interior information.

Two modifications are necessary. First, instead of filtering the Fourier coefficients directly, we pull out a leading radial power with one less that the full power for $m > 0$ so that the "normalized" coefficient vanishes at the magnetic axis. We then filter the "normalized" Fourier coefficient and then restore the radial factor. This helps to preserve the correct leading radial behavior near $\rho = 0$, which would be otherwise lost if the filtering were directly applied. Second, we modify the weighting of neighboring terms in the filter, Eq. (56), so that the highest frequency modes at $\rho = d\rho$ and at $\rho = 1 - d\rho$ are killed in one application of the filter. It should be further noted that care must be used when filtering divergence-free fields such as **B**, since the filtered fields are not necessarily divergence-free.

The filter requires as input to the PIES code three parameters: **fltrcnt**, which determines how many times the filter Eq. (56) is applied in succession to each Fourier coefficient during a given iteration; **s1**, which determines the amount of radial diffusion, and **s2** which determines the amount of radial antidiffusion. Default values used are **fltrcnt = 1, s1 = 1/2**, and **s2 = −1/2.**

V. Results

In this section we discuss the results of numerical experiments in which we apply the PIES code to two representative MHD equilibrium problems: an axisymmetric tokamak equilibrium (the analytic Solov'ev solution [29]) and a nonaxisymmetric finite β stellarator equilibrium first studied by Chodura and Schlüter [6]. In each case, we study how the convergence of the code depends on the numerical parameters of Table I, and on the physical data of Table II. We emphasize that since the PIES code has a high-dimensional numerical parameter space, it is difficult to characterize its properties with a moderate number of runs.

The convergence rate is measured by calculating the magnitude of the residual of the MHD equilibrium equations, Eqs. (1)–(3), at the end of each iteration. If we define the maximum norm of a Fourier series to be

$$\left\| \sum_{m,n} f_{m,n}(\rho) \sin(n\phi - m\theta) \right\| = \frac{1}{(m+1)(2n+1)} \max_{m,n,l} |f^{l}_{m,n}|, \tag{57}$$

then the residual is the maximum norm of the force balance expression, $\mathbf{J} \times \mathbf{B} - \nabla p$, normalized to the maximum pressure gradient (the latter is set to one for force-free equilibria):

$$\begin{aligned} \textbf{residual} = \frac{1}{\|p'(\rho)\|} \max\{ &\|\mathscr{J}^{\rho}(J^{\theta}B^{\phi} - J^{\phi}B^{\theta}) - p'(\rho)\|, \\ &\|\mathscr{J}^{\rho}(J^{\phi}B^{\rho} - J^{\rho}B^{\phi})\|, \|\mathscr{J}^{\rho}(J^{\rho}B^{\theta} - J^{\theta}B^{\rho})\| \}. \end{aligned} \tag{58}$$

Because the current density $\mathbf{J}(K)$ at the Kth iteration is determined from $\mathbf{B}(K)$ by force balance, the new magnetic field $\mathbf{B}(K+1)$ from Eq. (47) must be used in this expression for the residual to give useful information.

Numerical experiments similar to those described below showed that the convergence of the PIES code was consistent with various assumed approximations. Thus fixing all parameters and varying L showed that the code converges quadratically as $1/L^2$ at each iteration; this is consistent with the second-order radial discretizations discussed in Section III.B. Similarly fixing all parameters and varying either $\mathcal{M}$ or $\mathcal{N}$ gave approximately exponential convergence of the residual, again consistent with the assumed expansion of smooth functions as Fourier series. This is discussed in detail further below.

These convergence rates do not indicate how the computational effort of the PIES code scales with numerical parameters, which we briefly summarize here. The most expensive parts of the code are evaluation of the Fourier sums, Eqs. (8)–(9), when interpolating the magnetic field at general points in the domain (this is required when integrating the field line equations, Eq. (29)), and solving the Poisson equation, Eq. (5).

Interpolation of $\mathbf{B}$ involves an effort $\propto (L+1)(\mathcal{M}+1)(2\mathcal{N}+1)$, since a Fourier series similar to Eq. (8) must be evaluated for each of the $L+1$ field lines. Solving the Poisson equation [14] requires solution of a $(L+1)\times(L+1)$ block tridiagonal matrix with blocks of size $(\mathcal{M}+1)(2\mathcal{N}+1)\times(\mathcal{M}+1)(2\mathcal{N}+1)$ so the effort is $\propto (L+1)(\mathcal{M}+1)^3(2\mathcal{N}+1)^3$ if Gaussian elimination is used for the blocks. Although the effort for the Poisson solver grows rapidly with increasing angular resolution, in all the examples studied the large number of magnetic field interpolations dominated each iteration.

A feeling for the efficiency of the PIES code is given by considering the 5 period, aspect ratio 10, $l=2$ stellarator equilibrium considered below. For $\mathcal{M}=6$, $\mathcal{N}=3$, and $L=20$ there are about 50 Fourier modes and 20 radial grid surfaces. Four iterations were required to converge to a residual smaller than 10^{-3}, and these four iterations took about 2.5 min on a Cray 1. Over a large range in L, $\mathcal{M}$, and $\mathcal{N}$, the CPU time is simply proportional to these parameters since the field interpolation dominates. Thus twice as many radial grids or twice as many poloidal modes would require roughly twice as much CPU time.

Overall convergence of the code is determined by the crucial parameter **ftprec**, which effectively determines how many times magnetic field lines must be followed to obtain accurate Fourier coefficients in near-magnetic coordinates [22]. Fortunately, the computational effort grows only logarithmically with **ftprec**, since the Fourier coefficients are decreasing exponentially in magnitude with increasing $\mathcal{M}$ and $\mathcal{N}$.

V.A. *Axisymmetric Tokamak Equilibrium*

A test that is commonly used to verify equilibrium codes is the Solov'ev analytic tokamak equilibrium [29]. An axisymmetric field can be written in the form

$$\mathbf{B} = \nabla\psi_p \times \nabla\phi + g\,\nabla\phi, \tag{59}$$

where $g(\psi_p)$ is the poloidal current and ψ_p is the poloidal flux enclosed by a given magnetic surface. This satisfies the MHD equations, Eqs. (1)–(3), provided ψ_p satisfies the Grad–Shafranov equation [30]. In cylindrical coordinates (r, z, ϕ), an analytic solution is given by

$$\psi_p(r,z)=\psi_0\left(\frac{r}{R}\right)^2\left[2-\left(\frac{r}{R}\right)^2-4\alpha^2\left(\frac{z}{R}\right)^2\right], \tag{60}$$

with

$$p'(\psi)=\frac{8(1+\alpha^2)\,\psi_0}{R^4}=\text{constant}, \tag{61}$$

$$g=g_0=\text{constant}, \tag{62}$$

where R is the major radius of the torus and g_0, ψ_0, and α are arbitrary parameters.

These parameters can be chosen as follows. If we normalize $\mathbf{B}\cdot\nabla\phi=1$, the [redacted] choose a value $\alpha=1$ that gives nearly circular surfaces. The parameter ψ_0 determines the rotational transform near the magnetic axis. We have set $\psi_0=0.14$ to obtain an axis transform of about 0.14. The poloidal resolution is fixed at $\mathscr{M}=10$. This is effectively infinite poloidal resolution, since increasing $\mathscr{M}$ does not change the results. The convergence properties of the code are then studied as functions of the radial resolution L and of the Fourier coefficient precision **ftprec**.

We first test the PIES code by starting the code with the exact analytic solution, Eqs. (59) and (60), and with the profiles in Eqs. (61) and (62). The residual after one iteration then measures how accurately the code can approximate the analytic answer for given radial and angular resolution and also determines the overall convergence rate of the code (i.e., the constant coefficient of the error term arising from the finite radial discretization). We see from Fig. 3 that the code achieves quadratic convergence, i.e., that residual $\propto 1/L^2$ for sufficiently large L, with a constant of proportionality of about 0.1. This is a strong check that various finite difference approximations were consistently implemented. The figure also demonstrates that the code is indeed solving the MHD equilibrium equations, i.e., as the resolution increases, the initial exact solution is a fixed point of the discrete equations to better and better accuracy.

We next test the PIES code by perturbing the Solov'ev equilibrium and studying the convergence back towards the equilibrium solution. We add a perturbation of the form

$$\psi_1=\begin{cases}\varepsilon\rho\cos(\theta)(\rho_0^2-\rho^2) & \text{for} \quad \rho<\rho_0\\ 0 & \text{for} \quad \rho\geqslant\rho_0,\end{cases} \tag{63}$$

to Eq. (60), where the parameters ε and ρ_0 determine the magnitude of the outward shift of the magnetic axis and the radial limit of the perturbation, respectively.

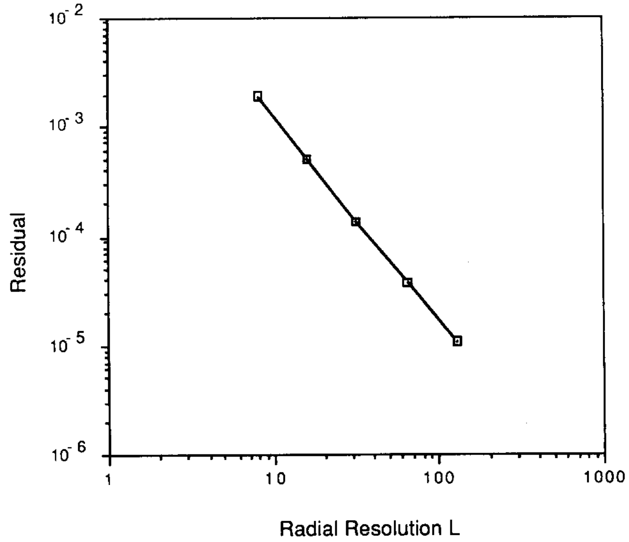

FIG. 3. Residual versus radial resolution L on a log–log plot. The residual is calculated after one iteration of the PIES code, starting with the analytic equilibrium, Eq. (60). The straight line was obtained by a least squares fit, giving an approximate quadratic dependence **residual** $\simeq 0.1\ L^{-1.9}$.

We have chosen values $\varepsilon = 0.3$ and $\rho_0 = 0.75$. We start the code with the perturbed **B** field and with the same profiles as before and iterate until no further changes occur or until the code diverges (the residual starts increasing). Figure 4 shows the (ρ, θ) grid for the perturbed initial state (a) and the final equilibrium state (b).

A more quantitative understanding of the convergence properties of the code is obtained by systematically varying numerical parameters that affect convergence, the most important of which relate to blending and filtering (see Section IV.E). In Fig. 5, we show how the residual varies with iteration number K for fixed angular and radial resolutions and with different combinations of fields being blended (see Eq. (51)). Figure 5a has no filtering of Fourier coefficients ($s1 = s2 = 0$), while for Fig. 5b, $s1 = \frac{1}{2}$ and $s2 = -\frac{1}{2}$.

We see from Figs. 5a and b that convergence is slow or not attainable unless the rotational transform $\iota(\rho)$ is blended. In the absence of filtering (Fig. 5a), blending ι alone is not sufficient for convergence. The code stops after two iterations because of numerical noise near the magnetic axis. We also learn from this figure that filtering has only a moderate affect on convergence, stabilizing the case where only ι is blended (–i–) and destabilizing the case where no fields are blended (– – –), and otherwise not affecting the convergence rate. Filtering has affected the attainable residual, presumably because the strong filtering used here has decreased the effective radial resolution. (The effect of radial resolution on the converged residual will be further discussed below.) Other combinations of blending were also tried (e.g., b– –, b–x, bi–, and –ix) and were found to give results similar to the curves in Fig. 5,

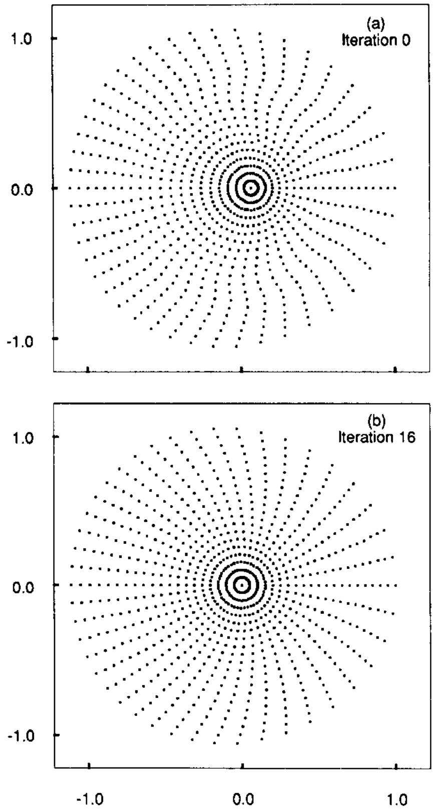

FIG. 4. (ρ, θ) grid for the 0th (Part a) and 16th (Part b) iterations of the perturbed Solov'ev equilibrium, Eq. (63). There is a Shafranov shift of about 10%. The final state, (b), closely approximates the analytic Solov'ev solution.

i.e., they confirmed the importance of always blending at least the rotational transform.

The peculiar importance of blending $\iota(\rho)$ can be understood by a more careful examination of output from the code. It is found that when the rotational transform ι is not blended, the transform is oscillating while the residual remains roughly constant (blending of ι would then be expected to damp the oscillations). The origin of the oscillations in ι is not hard to determine. The change $\delta\iota(K) = \iota(K) - \iota(K-1)$ in transform from one iteration to the next can be seen to satisfy $\delta\iota(K+1) \propto C/\delta\iota(K)$, with the constant of proportionality $C \simeq 1$. Thus increases in $\delta\iota$ in one iteration lead to decrease in the next, and oscillation tends to result. The derivation holds for equilibria with net toroidal current and so oscillations are predicted not to occur for stellarator equilibria with zero net toroidal current, as is confirmed by our calculations below.

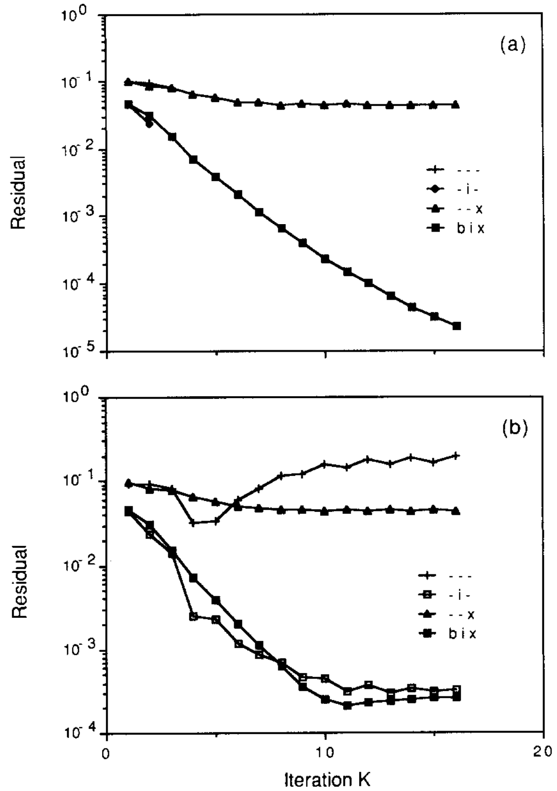

FIG. 5. Residuals (Eqs. (58)) versus iteration number K for the perturbed Solov'ev equilibrium, for $\mathcal{M} = 10$, $\mathcal{N} = 0$, $L = 20$, and **ftprec** $= 10^{-5}$. In (a), no filtering is used ($s1 = s2 = 0$), while in (b), $s1 = \frac{1}{2}$ and $s2 = -\frac{1}{2}$. Residuals are given when no fields are blended (---), when the rotational transform only is blended (-i-), when the coordinate x is blended (--x), and when all three fields are blended (bix). The blending parameter is $\frac{1}{2}$ in all cases. Curves that stop before the 17th iteration diverged because of numerical instabilities.

The effects of the radial resolution L and the finite Fourier precision **ftprec** are more carefully examined in Fig. 6, where the value of the residual achieved after 20 iterations is plotted as a function of radial resolution L for several different values of **ftprec**. For large values of **ftprec**, the residual is independent of L (for the range of L studied here) since the discretization error, of order $1/L^2$, is much smaller than **ftprec**. As **ftprec** gets significantly smaller than the radial discretization error, one sees quadratic convergence in the residual, until the latter is again of order **ftprec**. The quadratic error term has approximately the same coefficient (0.1) as that observed in Fig. 3.

The changes in the radial component of the magnetic field in the Solov'ev equilibrium is given in Fig. 7. The quantity B^ρ/B^ϕ measures the deviation of ρ from the magnetic surfaces. In the 0th iteration, B^ρ deviates from zero because of the pertur-

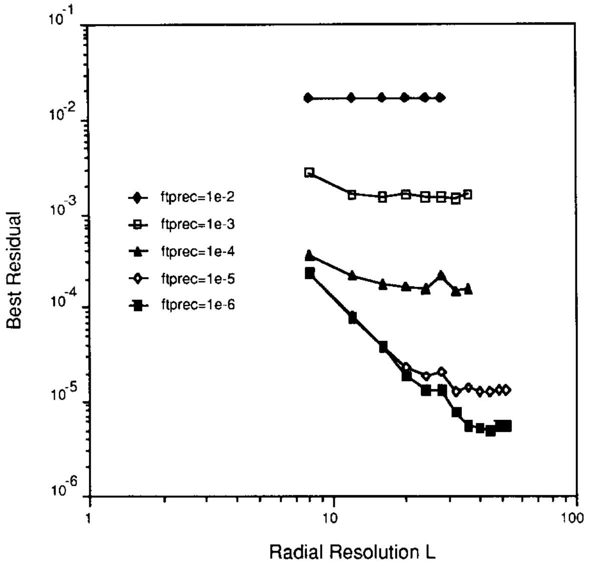

FIG. 6. Plot of residual versus radial resolution L for a fixed iteration number of twenty iterations, and for several different values of the Fourier precision parameter **ftprec**. The latter is seen to provide a cutoff to the residual, which cannot get smaller than about this amount. For the smallest values of **ftprec**, the residual is converging quadratically with approximately the same constant and exponent as given in Fig. 3, until the cutoff of **ftprec** is reached.

bation. Upon convergence, self-consistency has been attained to much better than the parameter **ftprec** $= 10^{-5}$. The random structure of the Fourier modes in Fig. 7b indicates that convergence has been attained to within the available number of significant digits (**ftprec**), so that only numerical noise remains.

For tokamak equilibria, we conclude that some blending (but not filtering) of fields is needed to obtain convergence, and that convergence is quadratic for sufficiently small **ftprec**. The code is unstable without blending.

V.B. *Nonaxisymmetric Stellarator Equilibrium*

The second example that we systematically studied was a more challenging nonaxisymmetric stellarator equilibrium. In the absence of axisymmetry, one expects resonances and island formation on rational magnetic surfaces. In practice, one can choose physical parameters (corresponding to a small value of the rotational transform) so that, with low to moderate numerical resolution, resonances do not occur. As the angular resolution increases ($\mathcal{M} \to \infty$, $\mathcal{N} \to \infty$), resonant denominators will appear and one must treat the magnetic islands and stochastic regions directly as discussed in Ref. [8, 22].

We chose a 5 period ($N = 5$), $l = 2$, aspect ratio = 10 stellarator, corresponding to the parameters of the WVIIA stellarator. As is appropriate for stellarators, we

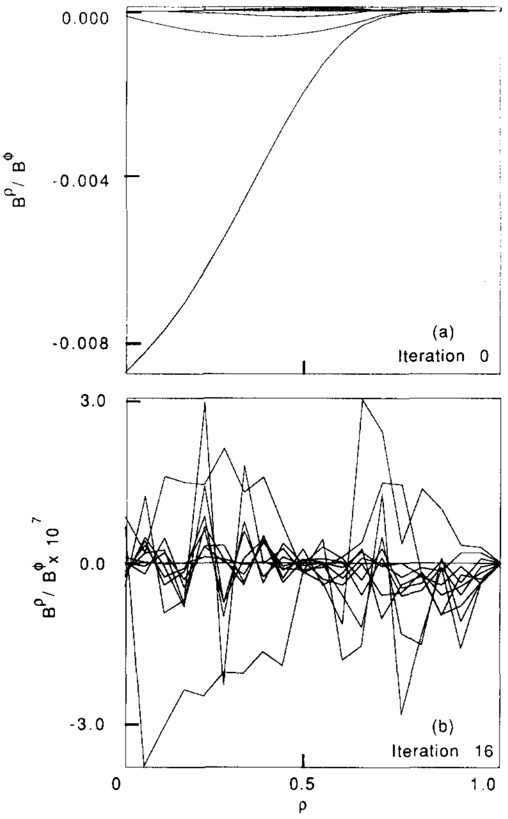

FIG. 7. Initial (a) and final (b) plots of the Fourier coefficients of B^ρ/B^ϕ versus ρ for the perturbed Solov'ev equilibrium.

imposed zero net toroidal current on each magnetic surface ($I' = 0$). We assumed a pressure profile of the form $p = p_0(1-\rho^2)^2$ and a value of $p_0 = 5.4 \times 10^{-3}$, to give a Shafranov shift of about 20% of the minor radius. A Bessel function solution for the vacuum field in a cylinder was used as an initial guess for the field, $\mathbf{B}^0 = \nabla\chi$ with

$$\chi(\rho, \theta, \phi) = b_0\phi + b_2 I_2(N\rho/R)\sin(2\theta - N\phi), \tag{64}$$

where $I_2(\rho)$ is a modified Bessel function. Taking $b_2/b_0 = 0.667$ gives a relatively low rotational transform of about $\iota = 0.14$ at the magnetic axis and also yields a relatively low shear. This avoids resonances for moderate angular resolutions.

The convergence properties of the stellarator equilibrium as a function of L and **ftprec** were found to be very similar to those for the Solov'ev equilibrium described in the previous section, with quadratic convergence being achieved with increasing L. Memory restrictions on the CRAY-XMP prevented extensive tests of convergence with respect to poloidal and toroidal mode numbers, but various runs indicated an exponential decrease in residual with increasing poloidal number $\mathcal{M}$, as expected for spectral representations of smooth functions. For $L=24$, $\mathcal{M}=6$, and $\mathcal{N}=3$, a smallest residual of 8.2×10^{-4} was obtained, while doubling the resolution to $\mathcal{M}=12$ and $\mathcal{N}=6$ with L fixed gave a smallest residual of 6.4×10^{-5}. The factor of 13 improvement is consistent with spectral (exponential) convergence.

For the rest of this section, we fix $L=20$ and **ftprec** $=10^{-5}$. We also take $\mathcal{M}=6$ and $\mathcal{N}=3$. Figure 8 shows the (ρ, θ) grid for the initial and converged magnetic field. A Shafranov shift of about 20% is apparent in the converged field. Both the axis shift and the shape of the magnetic surfaces are in reasonable agreement with the previous calculation of Chodura and Schlüter [6].

When the above initial conditions were used without blending or filtering, the residual and corrections were found to decrease and then increase with successive iterations, indicating some numerical instability of the algorithm. Figure 9 is similar

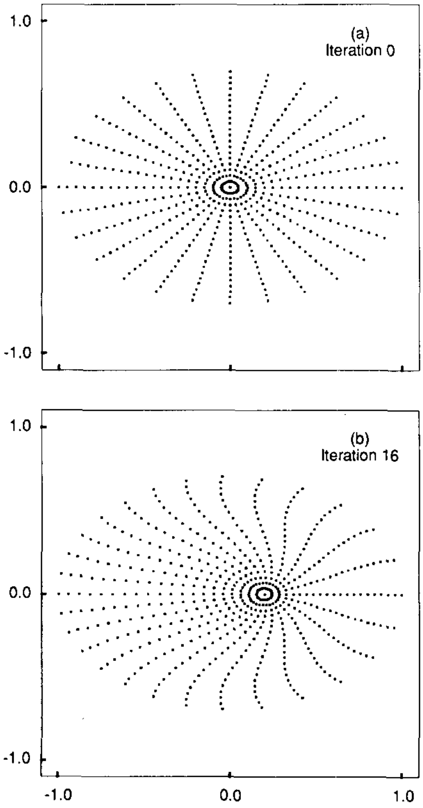

FIG. 8. The (ρ, θ) grid in the $\phi=0$ plane for the initial (a) and final (b) states of the 5 period stellarator field with $l=2$, aspect ratio $=10$, and $\beta=5.4\times10^{-3}$. Numerical parameters are $L=20$, $\mathcal{M}=6$, $\mathcal{N}=3$, **ftprec** $=10^{-5}$. The figure is in good agreement with an earlier calculation of Chodura and Schlüter [6].

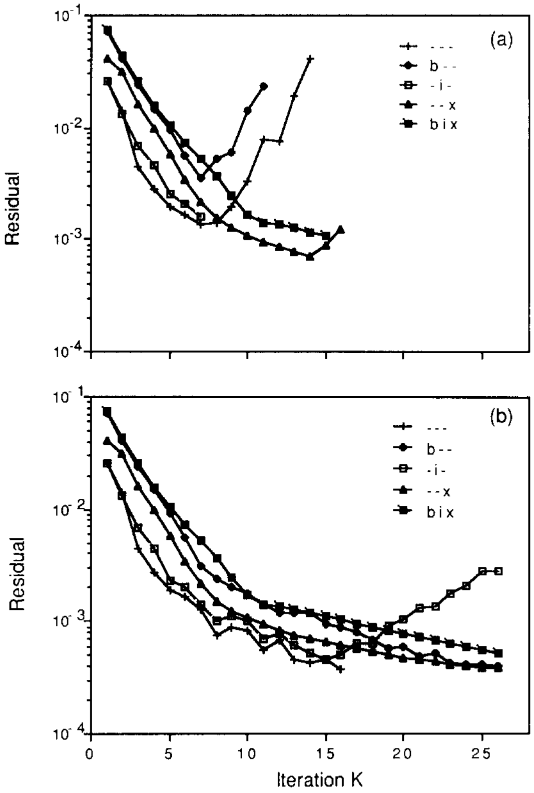

FIG. 9. Residual versus iteration number K for the low-shear stellarator equilibrium. In (a), no filtering is used ($s1 = s2 = 0$), while in (b), $s1 = \frac{1}{2}$ and $s2 = -\frac{1}{2}$. The blending parameter is $\frac{1}{2}$ in all cases.

to Fig. 5 in showing how convergence depends on filtering and blending, with (a) giving the results without filtering and (b) with filtering. The results contrast sharply with the tokamak runs: convergence is not attained for any combination of blending unless filtering is used (the default parameters of $s1 = \frac{1}{2}$ and $s2 = -\frac{1}{2}$ were chosen for part (b)). Even when filtering of Fourier coefficients is turned on, not all combinations of fields being blended result in convergence. The runs that do converge seem stable out to the largest iteration numbers studied (about 30). The smallest residual attained for convergent runs again depends on the parameter **ftprec**, approaching but never becoming smaller than this value.

Because of the importance of filtering in stellarator runs, we studied more systematically the sensitivity of convergence to the filtering parameters $s1$ and $s2$ in Fig. 10. The main conclusion of this figure is the need for some minimal radial smoothing. It is not necessary to entirely remove the highest frequency component ($s1 = \frac{1}{4}$). When complete filtering of the highest frequency component ($s1 = \frac{1}{2}$) is employed, a subsequent antidiffusion improves the accuracy of the solution somewhat by restoring the amplitudes of the lower frequency modes.

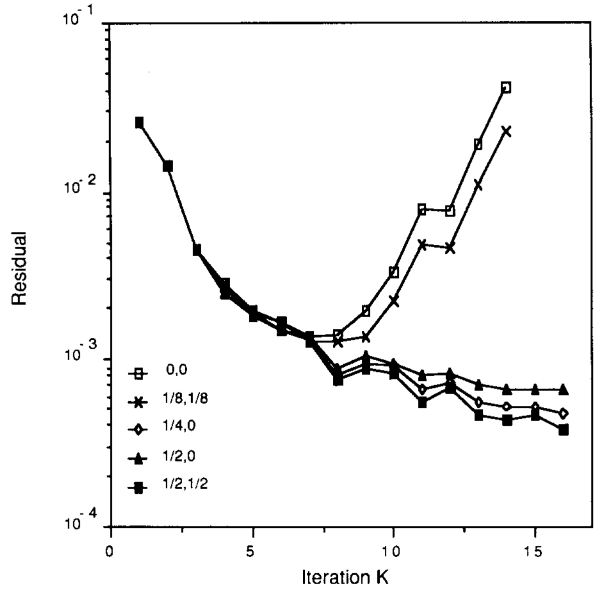

FIG. 10 Residual versus iteration number K for the low-shear stellarator equilibrium, for no blending and different choices of filtering parameters s_1 and s_2 (see Eqs. (55) and (56)). The curves are labeled by ordered pairs ($s1$, $-s2$).

With insufficient filtering, the code stops because of numerical noise appearing in Fourier coefficients near the magnetic axis. This prevents the algorithm used for transforming to near-magnetic coordinates from finding a magnetic axis [22]. Some insight into this instability is given by Fig. 11, which shows the radial dependence of the Fourier coefficients of B^{ρ}/B^{ϕ} at the end of several iterations. The Fourier coefficients are smooth and well behaved until about the fifth iteration, at which point kinks with magnitude above **ftprec** appear near the magnetic axis ($\rho \leqslant 0.3$). These kinks continue to grow in magnitude and remain localized in the vicinity of the magnetic axis. Eventually, they exceed all other modes in magnitude and the code fails because it cannot find a magnetic axis.

Numerical experiments showed that this instability is a subtle one, with its growth rate depending sensitively on the discretization used in the Poisson solver, and not being sensitive at all to numerical details of integrations along field lines and the transformation to near-magnetic coordinates. This situation is similar to some experiences with 3-dimensional variational codes in flux coordinates, for which subtle discretizations were also needed to obtain numerically stable codes [15].

Some examples of runs we made to explore this instability are given in Fig. 12. It might be expected that subtracting off $\nabla \times \mathbf{B}$ from Ampère's law and solving for the correction to the new magnetic field, rather than for the field itself, would

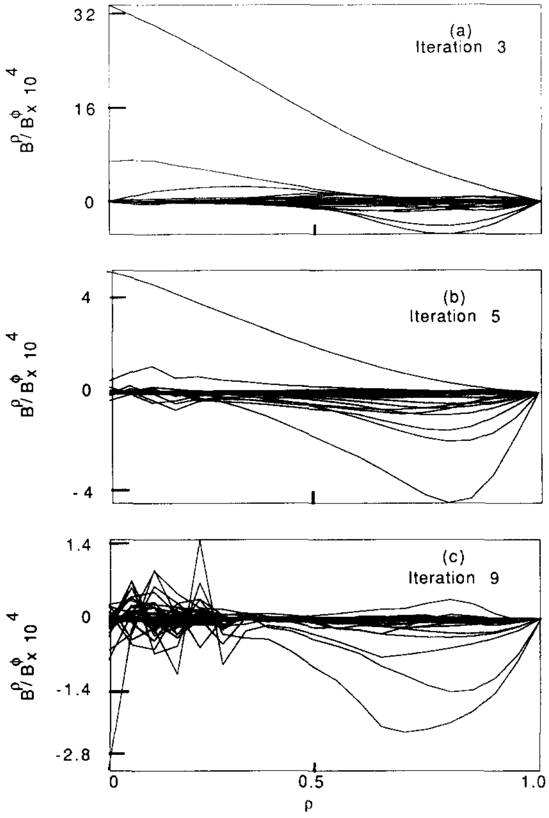

FIG. 11. Fourier coefficients of B^ρ/B^ϕ as functions of ρ during successive iterations of the PIES code in the absence of radial filtering. A slowly growing instability near the magnetic axis eventually dominates the Fourier modes, preventing convergence.

improve convergences (see Eq. (46)). In particular, this reduces the large numerical cancellation among the terms that make up B^ρ, which is tending to zero during convergence. The figure shows that this modification had little effect on the initial convergence rate of the code and was actually destabilizing for long runs.

Figure 12 also shows that switching from the generalized Boozer gauge, Eq. (42), to another gauge, Eq. (45), had little influence on the instability, even though these gauges represent different numerical approximations. In this case, the convergence rate for the two gauges was nearly the same, with the final residual being slightly smaller for the Boozer gauge. We have found that the gauge of Eq. (45) can be considerably more accurate than the Boozer gauge for axisymmetric equilibria with nearly circular flux surfaces.

The above results may be summarized by stating that—for stellarator equilibria with magnetic surfaces and with zero net toroidal current—the nonvariational algorithm is numerically unstable unless some filtering *and* blending of Fourier

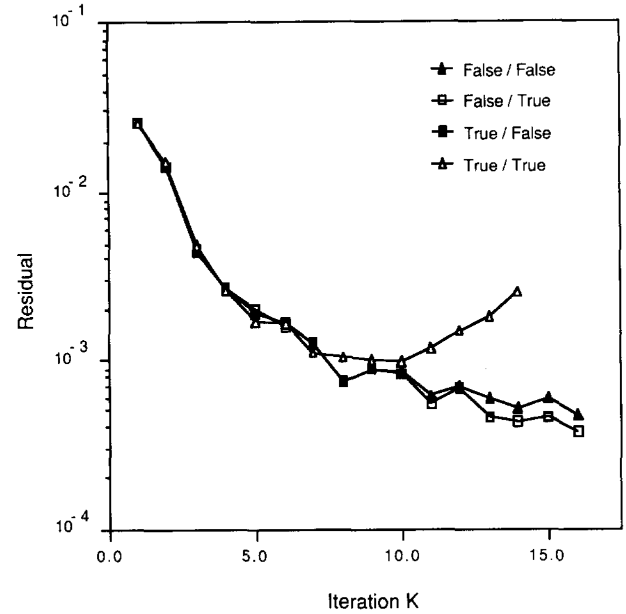

FIG. 12. Residual versus iteration number K for different choices of the flags **rmcrlf** and **bzrggf**. The former flag if **true** subtracts off $\nabla \times \mathbf{B}$ when solving Ampère's law as in Eq. (46); if **false**, no subtraction is performed. Similarly, if **bzrggf** is **true**, the expression Eq. (43) is used to evaluate $\mathbf{h}$ in terms of $\mathbf{J}$, otherwise Eq. (45) is used.

coefficients is used. Filtering and blending suffices to stabilize the algorithm and to obtain convergence of O(**ftprec**), the precision of Fourier coefficients after transformation to near-magnetic coordinates.

VI. CONCLUSIONS

In this paper, we have discussed in detail a new 3D MHD equilibrium code and have shown it to converge rapidly and efficiently for representative tokamak and stellarator equilibria that have magnetic surfaces. The convergence properties of the code were studied for various physical and numerical parameters; only a small portion of the high-dimensional parameter space could be studied.

The importance of the code lies in its use of a nonvariational algorithm and in the formulation of the MHD equilibrium equations in terms of a magnetic differential equation (Eq. (35) above). Both elements make this approach more suitable for calculating MHD equilibria that lack magnetic surfaces [12]. This approach is also of interest in its own right for equilibria with surfaces since so few alternatives to variational algorithms have been explored for 3-dimensional plasmas.

One important conclusion obtained from our numerical experiments on representative toroidal equilibria is that the PIES algorithm converges. Although the basic algorithm was already considered in the 1950's, our numerical results are the first to show that this algorithm is generally convergent in three dimensions.

A second conclusion is that the PIES algorithm converges rather rapidly. Typically of order ten iterations leads to an equilibrium that has the smallest possible residual that is consistent with the parameters **ftprec**, $\mathscr{M}$, $\mathscr{N}$, and L. This is several orders of magnitude fewer iterations than are typically required for variational codes, in which many (but computationally less demanding) steps are taken at each iteration towards an extremum of the variational functional. The rapid convergence shows that there are no vestiges of any fast time scale behavior in the algorithm.

Each iteration is relatively efficient. Although only a modest effort was made to vectorize the PIES code for running on a CRAY supercomputer and the most efficient algorithms possible were not used, the code is already competitive with other 3-dimensional equilibrium codes in that only a few CRAY-XMP minutes are need to attain a stellarator equilibrium to four significant digits. For many purposes, this is an acceptable time although further improvements in speed would be useful. The scaling properties of the different parts of the code are well understood and can be used to estimate the CPU time of a run with given radial and angular resolutions.

Further improvements could be made by using a more efficient algorithm for constructing near-magnetic coordinates, by using a more carefully vectorized code for solving block tridiagonal matrices, and by using optimized coordinates [23] to reduce the size of the matrix associated with solving Ampère's law. The code also lends iftself to parallel computer architectures since the time consuming part of the code, following $L+1$ field lines to transform quantities to near-magnetic coordinates, can be done in parallel and independently on each magnetic surface.

A third conclusion is that the PIES algorithm, although convergent, sometimes requires application of blending or of a radial filter to remove high frequency modes (a form of radial dealiasing). In the absence of this filtering, high-frequency noise in various Fourier coefficients appears and grows after many iterations. Similar instabilities are known in meteorological codes [16], and, more generally, in many nonlinear codes and are treated in a similar and successful manner.

An analytical understanding of this instability is presently lacking because of the difficult nonlinear character of the PIES algorithm. The above numerical experiments have shown that the instability is insensitive to many details of the PIES algorithm, e.g., whether finite differences or splines are used to differentiate the coordinates to get the metric elements, or whether one subtracts $\nabla \times \mathbf{B}$ from the right side of Ampère's law to avoid cancellations between large terms, or the choice of gauge in representing the solution to Ampère's law. Instead, the instability is related to how well one approximates the correct asymptotic analytic behavior of Fourier coefficients near the magnetic axis. Stability was obtained by our explicit treatment of leading radial powers when differentiating or averaging Fourier coefficients.

It is our hope that the results presented here will stimulate further consideration of nonvariational algorithms and their applications to 3-dimensional MHD equilibria.

APPENDIX A: THE OLD **b** FIELD IN THE NEW MAGNETIC COORDINATES

In generalizing the code to handle magnetic islands with net toroidal current or to use magnetic fields from previous iterations to improve convergence (see Eq. (46)), it is useful to express the Fourier coefficients of a magnetic field in terms of the magnetic coordinates defined by its field lines. This can be done as follows.

At the end of one iteration, the new magnetic field, Eq. (47),

$$\mathbf{b} = (b^{\rho}(\rho, \theta, \phi), b^{\theta}(\rho, \theta, \phi), b^{\phi}(\rho, \theta, \phi)), \tag{A1}$$

is known in terms of the magnetic coordinates (ρ, θ, ϕ) of the field from the previous iteration, **B**. The field lines of **b** define a new magnetic coordinate system, $(\bar{\rho}, \bar{\theta}, \phi)$ and we would like to obtain the Fourier expansion of the quantities

$$\mathbf{b} \cdot \mathbf{e}_{\bar{\rho}} = b_{\bar{\rho}}(\bar{\rho}, \bar{\theta}, \phi), \qquad \mathbf{b} \cdot \mathbf{e}_{\bar{\theta}} = b_{\bar{\theta}}(\bar{\rho}, \bar{\theta}, \phi), \qquad \mathbf{b} \cdot \mathbf{e}_{\phi} = b_{\phi}(\bar{\rho}, \bar{\theta}, \phi), \tag{A2}$$

in these new magnetic coordinates.

The first step is to transform Eq. (A1) to cylindrical coordinates x and y:

$$\begin{aligned} b^x(\rho, \theta, \phi) &= \mathbf{b} \cdot \nabla x = \mathbf{b} \cdot (x_{,\rho} \nabla\rho + x_{,\theta} \nabla\theta + x_{,\phi} \nabla\phi) \\ &= b^{\rho} x_{,\rho} + b^{\theta} x_{,\theta} + b^{\phi} x_{,\phi} \\ b^y(\rho, \theta, \phi) &= b^{\rho} y_{,\rho} + b^{\theta} y_{,\theta} + b^{\phi} y_{,\phi}. \end{aligned}$$

We then evaluate b^x and b^y along field lines of **b**, applying the algorithm of Ref. [22]. This is done in parallel with the usual scalar fields (x, y, and b^{ϕ}), so that the overhead is small and vectorization is achieved. This gives the Fourier expansion of b^x and b^y in the new magnetic coordinates, $(\bar{\rho}, \bar{\theta}, \phi)$.

Next, we form the covariant components of **b** in the new magnetic coordinates by appropriate dot products:

$$\begin{aligned} b_{\bar{\rho}}(\bar{\rho}, \bar{\theta}, \phi) &= \mathbf{b} \cdot \mathbf{x}_{,\bar{\rho}} = \mathbf{b} \cdot (x_{,\bar{\rho}} \hat{\mathbf{x}} + y_{,\bar{\rho}} \hat{\mathbf{y}}) = b^x x_{,\bar{\rho}} + b^y y_{,\bar{\rho}} \\ b_{\bar{\theta}}(\bar{\rho}, \bar{\theta}, \phi) &= b^x x_{,\bar{\theta}} + b^y y_{,\bar{\theta}}. \\ b_{\phi}(\bar{\rho}, \bar{\theta}, \phi) &= \begin{cases} b^x x_{,\phi} + b^y y_{,\phi} + (R + x)^2 b^{\phi} & \text{finite aspect ratio,} \\ b^x x_{,\phi} + b^y y_{,\phi} + R^2 b^{\phi} & \text{infinite aspect ratio.} \end{cases} \end{aligned}$$

Using $\nabla\varphi = \hat{\varphi}/(R + x)$. Since b^{ϕ} is also evaluated along field lines, all quantities on the right side are known Fourier series in the latest magnetic coordinates, $(\bar{\rho}, \bar{\theta}, \phi)$. This gives the required expressions, Eq. (A2).

ACKNOWLEDGMENTS

We thank A. Boozer and D. Monticello for numerous helpful conversations. We also thank O. Buneman for making available to us his Fast Hartley Transform routines. Chris Kerr for discussions of filtering routines used by meteorologists, and John Johnson for a critical reading of an earlier version of this paper. This work was supported by the U.S. Department of Energy Contract DE-AC02-76-CHO-3073.

REFERENCES

1. A. SCHLÜTER AND U. SCHWENN, *Comput. Phys. Commun.* **24**, 263 (1981).
2. F. BAUER, O. BETANCOURT, AND P. GARABEDIAN, *A Computational Method in Plasma Physics* (Springer-Verlag New York, 1982), 2nd ed.
3. U. SCHWENN, *Comput. Phys. Commun.* **31**, 167 (1984).
4. L. L. LAO, J. M. GREENE, T. S. WANG, F. J. HELTON, AND E. M. ZAWADZKI, *Phys. Fluids* **28**, 869 (1985).
5. S. P. HIRSHMAN AND D. K. LEE, *Comput. Phys. Commun.* **39**, 161 (1986).
6. R. CHODURA AND A. SCHLÜTER, *J. Comput. Phys.* **41**, 68 (1981).
7. T. C. HENDER, B. A. CARRERAS, L. GARCIA, J. A. ROME, AND V. E. LYNCH, *J. Comput. Phys.* **60**, 76 (1985).
8. A. H. REIMAN AND H. S. GREENSIDE, *Comput. Phys. Commun.* **43**, 157 (1986).
9. L. SPITZER, *Phys. Fluids* **1**, 253 (1958).
10. H. GRAD AND H. RUBIN, in *Proceedings, Second United Nations International Conf. on the Peaceful Uses of Atomic Energy* (United Nations, Geneva) Vol. 31, p. 190.
11. A. H. BOOZER, *Phys. Fluids* **27** No. 8, 2110 (1984).
12. A. REIMAN AND A. BOOZER, *Phys. Fluids* **27** No. 10, 2446 (1984).
13. A. REIMAN AND H. GREENSIDE, Princeton Plasma Physics Laboratory Report PPPL-2579, Nov., 1988.
14. H. S. GREENSIDE AND A. H. REIMAN, Numerical solution of Ampère's law in three-dimensional toroidal domains, unpublished.
15. S. HIRSHMAN, private communication (1987).
16. F. MESINGER AND A. ARAKAWA, GARP Publications Series No. 17, Vol. I, August 1976.
17. J. KISSLINGER AND H. WÖBIG, 12th European Conference on Controlled Fusion and Plasma Physics, Budapest, Sept. 1985.
18. A. I. SHESTAKOV, *Comput. Phys. Commun.* **31**, 227 (1984).
19. L. M. DEGTYAREV, V. V. DROZDOV, M. I. MIKHAILOV, V. D. PUSTOVITOV, AND V. D. SHAFRANOV, *Soviet J. Plasma Phys.* **11**, 22 (1985).
20. S. C. CHANG AND J. J. ADAMCZYK, *J. Comput. Phys.* **60**, 23 (1985); S. C. CHANG AND J. J. ADAMCZYK, *J. Comput Phys.* **60**, 41 (1985); S. C. CHANG, *J. Comput. Phys.* **67**, 91 (1986).
21. M. D. KRUSKAL AND R. M. KULSRUD, *Phys. Fluids* **1**, 265 (1958).
22. A. H. REIMAN AND H. S. GREENSIDE, *J. Comput. Phys.* **75**, 423 (1988).
23. S. P. HIRSHMAN AND H. K. MEIER, *Phys. Fluids* **8**, 5 (1985).
24. L. S. SOLOV'EV AND V. D. SHAFRANOV, in *Review of Plasma Physics*, edited by M. A. Leontovich (Consultants Bureau, New York), Vol. V, 1970.
25. C. DEBOOR, *A Practical Guide to Splines*, (Springer-Verlag, New York, 1978).
26. P. GARABEDIAN, *Partial Differential Equations* (Wiley, New York, 1964).
27. D. A. SMITH, W. F. FORD, AND A. SIDI, *SIAM Rev.* **29**, No. 2, 199 (1987).
28. R. SHAPIRO, *Rev. Geophys. Space Phys.* **8**, 359 (1970).
29. L. S. SOLOV'EV, *Soviet Phys. JETP* **26**, No. 2, 400 (1968).
30. J. P. FREIDBERG, *Rev. Mod. Phys.* **54**, No. 3, 801 (1982).