JOURNAL OF COMPUTATIONAL PHYSICS 75, 423–443 (1988)

Numerical Solution of Three-Dimensional Magnetic Differential Equations

A. H. REIMAN AND H. S. GREENSIDE

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

Received January 30, 1987; revised May 15, 1987

A computer code is described that solves differential equations of the form $\mathbf{B}\cdot\nabla f = h$ for a single-valued solution f, given a toroidal three-dimensional divergence-free field $\mathbf{B}$ and a single-valued function h. The code uses a new algorithm that Fourier decomposes a given function in a set of flux coordinates in which the field lines are straight. The algorithm automatically adjusts the required integration lengths to compensate for proximity to low order rational surfaces. Applying this algorithm to the Cartesian coordinates defines a transformation to magnetic coordinates, in which the magnetic differential equation can be accurately solved. Our method is illustrated by calculating the Pfirsch–Schlüter currents for a stellarator. © 1988 Academic Press, Inc.

I. INTRODUCTION

Magnetic differential equations constitute an important class of problems in magnetohydrodynamics (MHD) [1, 2]. These are equations of the form

$$\mathbf{B}\cdot\nabla f = h, \tag{1}$$

where $\mathbf{B}$ is a given divergence-free vector field, h is a given source term, and f is a function to be determined. Such equations reflect the anisotropy of the physics in a conducting medium immersed in a strong magnetic field. Motion along the field lines of $\mathbf{B}$ is fundamentally different from motion transverse to these lines. Of particular interest for toroidal magnetic confinement of plasmas is the situation where a field line winds about indefinitely on a toroidal surface. The solution of Eq. (1) is then sensitive to small errors in the representation of the $\mathbf{B}\cdot\nabla$ operator, particularly near surfaces with low order rational winding number of the field line, $\iota\!\!\!-$. Accurate solution of Eq. (1) therefore poses a formidable numerical challenge.

In this paper we present a numerical technique, and its implementation as a computer code, for accurately solving magnetic differential equations in a toroidal geometry, i.e., for three-dimensional domains that are periodic in two spatial directions. Two important examples that we discuss are the equation for the pressure,

$$\mathbf{B}\cdot\nabla p = 0, \tag{2}$$

0021-9991/88 $3.00

and the equation for the stream-function of the current density, ν [2]

$$\mathbf{B}\cdot\nabla\nu = p' + (g' + \iota I')\,\mathbf{B}\cdot\nabla\phi, \tag{3}$$

where $p'(\Psi) = dp/d\Psi$ is the pressure profile, $g'(\Psi)$ is the poloidal current profile, $I'(\Psi)$ is the toroidal current profile, and ϕ is the toroidal angle. The immediate motivation for the work described in this paper was the need to solve these equations for a three-dimensional MHD equilibrium code [3].

Our approach for solving Eq. (1) is to construct numerically a transformation from Cartesian coordinates to a set of flux coordinates in which the field lines are straight ("magnetic coordinates"). A Fourier transform of Eq. (1) in these coordinates reduces it to a trivially soluble algebraic equation. The inverse of the coordinate transformation to magnetic coordinates then yields the solution in Cartesian coordinates. At the heart of the solution is an algorithm which constructs a Fourier decomposition in magnetic coordinates of any given function. The algorithm automatically adjusts the required integration lengths to compensate for proximity to low order rational surfaces. Fourier decomposition of the Cartesian coordinates defines the transformation to the magnetic coordinate system.

An algorithm for Fourier decomposing any given function in magnetic coordinates has been introduced for 3D transport applications [4, 5]. Some aspects of that algorithm are incompletely automated. In addition, no estimates of the accuracy of the numerical Fourier coefficients have been given in this work. Even if such estimates are obtained, the algorithm has no provision for improving the accuracy, let alone for obtaining a specified accuracy. These deficiencies did not pose major problems in the context of a Monte Carlo transport code, but are clearly undesirable. In the context of our work on three-dimensional MHD equilibrium, they are unacceptable. We have adapted some of the ideas in this earlier work to develop our algorithm.

In this paper we assume that the field lines of **B** lie in a set of nested flux surfaces. This is implicit in our assumption that a magnetic coordinate system exists. The assumption is appropriate, for example, for stellarator vacuum fields, which are designed to have negligibly small islands and stochastic regions. More general three-dimensional magnetic fields can have large islands and stochastic regions. Even in the general case, however, any toroidal volume of interest can be decomposed into a set of regions with nested flux surfaces plus a set of stochastic regions. (Regions whose size falls below our spatial resolution are ignored, so that the number of different regions that must be considered remains finite.) Physical quantities of interest are constant in the stochastic regions, so there is no need to solve Eq. (1) there.

We could, in principle, solve the general case by decomposing as just described and applying the algorithm discussed in this paper to each of the regions separately. In fact, it is also often physically reasonable to take the quantities of interest constant in the magnetic islands. For example, the growth of islands tends to flatten the current profile. If the quantities of interest are taken to be constant in the islands and the stochastic regions, it is only necessary to solve Eq. (1) in those regions where nested flux surfaces enclose the magnetic axis. We have recently made some progress in implementing such a procedure [3]. We will not discuss this more general case further in this paper and deal only with a single region of nested flux surfaces. The focus is on the details of our method for constructing Fourier decompositions in magnetic coordinates, dwelling particularly on the issues raised by the presence of rational surfaces.

The assumption of nested flux surfaces is equivalent to postulating the existence of a single valued function $\Psi(x, y, z)$ such that $\mathbf{B} \cdot \nabla\Psi = 0$. A coordinate system (Ψ, Θ, Φ), where Θ is a poloidal angle and Φ is a toroidal angle, is called a "flux coordinate system." For toroidal plasma confinement devices, Ψ is often taken to be the net toroidal (or poloidal) flux enclosed by a magnetic surface. Several three-dimensional MHD equilibrium codes currently use flux coordinates [6–8]. These codes express the magnetic field in terms of the coordinates and solve for the coordinates directly. In our case we are given the field and want to construct the corresponding coordinates. Our field could come, for example, from a vacuum field solver, or from a 3D equilibrium solver not written in flux coordinates.

In addition to requiring that the coordinates we construct be flux coordinates, we impose the condition that the magnetic field lines be straight in these coordinates [9, 10]. Coordinates which satisfy these two conditions are sometimes called "magnetic coordinates." Magnetic differential equations have a particularly tractable form in magnetic coordinates, as we will discuss in Section II.

The remainder of the paper consists of four sections. In Section II we describe our method for solving magnetic differential equations numerically. The method requires an algorithm for Fourier decomposing any given function in magnetic coordinates. That algorithm is presented in Section III. Numerical accuracy is discussed in Section IV, focusing particularly on the issues raised by the presence of low order rational surfaces. Finally, we present an illustrative solution of Eq. (3) for a stellarator vacuum field in Section V. At the end of the paper, for reference, we have collected together in one table the most important numerical parameters discussed (Appendix B).

II. Solution of Magnetic Differential Equations

To solve magnetic differential equations, we take advantage of the fact that Eq. (1) has a particularly tractable form in magnetic coordinates. In magnetic coordinates with the toroidal flux as radial coordinate, $\mathbf{B}$ has the canonical form

$$\mathbf{B} = \nabla\Psi \times \nabla\Theta + \iota \nabla\Phi \times \nabla\Psi, \tag{4}$$

where $\iota(\Psi)$ is the rotational transform (winding number or twist) of the magnetic field line on a given Ψ surface, and Θ and Φ are, respectively, poloidal and toroidal angles. Substituting Eq. (4) into Eq. (1) transforms Eq. (1) to the simple form

$$\left(\frac{\partial}{\partial\Phi}+\iota\frac{\partial}{\partial\Theta}\right)f=\mathscr{J}^{\Psi}h, \tag{5}$$

where $\mathscr{J}^{\Psi}=1/B^{\Phi}=1/\mathbf{B}\cdot\nabla\Phi$ is the Jacobian of the transformation from Cartesian to magnetic coordinates. Because the coefficients on the left-hand side of Eq. (5) are independent of the angles (that is, the field lines are straight in this coordinate system), Fourier transformation of the equation yields an algebraic equation for the Fourier coefficients of f in terms of the Fourier coefficients of the right-hand side. To determine f uniquely, a constant of integration must be specified on each flux surface, corresponding to the $n=0$, $m=0$ Fourier component of f (where n and m are, respectively, the toroidal and poloidal mode numbers). Note that for a more general flux coordinate system in which the field lines are not straight Fourier transformation would have led to a convolution and inversion of a matrix to obtain the solution.

Equation (4) does not uniquely specify the angles. We have found it convenient to choose Φ to be the uniform toroidal angle ϕ of a cylindrical coordinate system. Throughout the rest of the paper we will retain the notation that (Ψ, Θ, Φ) denotes a general coordinate system having toroidal flux as the radial coordinate, but with no assumptions about the toroidal angle, while (Ψ, Θ, ϕ) denotes a magnetic coordinate system having the uniform geometric toroidal coordinate.

The key to our numerical solution of Eq. (1) is an algorithm which constructs a Fourier decomposition of any given function in magnetic coordinates. The Fourier components of $\mathscr{J}^{\Psi}h$ determine those of f, and therefore determine the solution f as a function of Θ and ϕ on each flux surface. The Fourier decomposition of the Cartesian coordinates x, y, and z specifies their Fourier components as a function of Θ and ϕ on each flux surface, thus determining the transformation $\mathbf{x}(\Psi, \Theta, \phi)$.

To Fourier decompose numerically in magnetic coordinates, we make use of an insight due to Boozer [4]. He observed that if a magnetic field line covers a good flux surface, the one-dimensional Fourier decomposition of any function along the field line contains sufficient information to construct the two-dimensional Fourier representation of that function on the flux surface in a coordinate system where the field lines are straight. The difficult part of Boozer's scheme is the identification of the two-dimensional Fourier modes with those in the one-dimensional Fourier decomposition of a function along a given field line. While this can be done by hand [5], almost any practical application of a code to solve Eq. (1) requires an automated method for making this identification. In Section III, we present a completely automated procedure for calculating the Fourier representation of any function on the magnetic surfaces of $\mathbf{B}$. The crucial step is an iterative algorithm for the precise determination of the rotational transform that requires a minimum of integration along field lines. Knowledge of ι allows us to identify directly the discrete spectral lines of the Fourier transform along a field line with the corresponding spectral lines in the two-dimensional Fourier transform on the magnetic surface.

In following field lines, the value of the magnetic field is required throughout the toroidal domain. For typical applications, the field is known only on a grid, and some interpolation scheme is required. Even when the field is known everywhere in principle, as when the field is calculated by the Biot-Savart law from a set of coils, it may be more efficient to calculate the field first at a discrete set of points and then to interpolate values elsewhere from these points. This is done in the Gourdon code [11] for example, where the magnetic field is first calculated on a rectangular grid and then linearly interpolated to intermediate points. Our code uses an exponentially accurate (Fourier) interpolation in the poloidal and toroidal coordinates, and a second-order accurate interpolation in the radial coordinate. The details are described at the end of Section III.

In this paper we make the simplifying assumption that the magnetic field is symmetric with respect to double reflection through an appropriate poloidal and equatorial plane. This is often the case for magnetic fields of interest. For example, stellarator fields almost always have this symmetry. Generalization of our algorithm to the nonsymmetric case is straightforward. Only our method for finding the magnetic axis would have to be modified, as discussed in Section III.

To solve Eq. (1) numerically throughout some given toroidal region, we follow magnetic field lines on a set of flux surfaces which provide a radial coordinate grid for the region. It is convenient to define a radial coordinate $\rho(\Psi)$ which is uniformly spaced between these surfaces. We calculate $\Psi(\rho)$ from the ratio of the Jacobians between Cartesian and flux coordinates with ρ and Ψ as independent variables, respectively. Fourier decomposition of the Cartesian coordinates determines $\mathbf{x}(\rho, \Theta, \phi)$. By numerical differentiation, we calculate the covariant basis vectors and then the Jacobian for the flux coordinates with ρ as independent variable,

$$\mathscr{J}^{\rho}=\frac{\partial(x, y, z)}{\partial(\rho, \Theta, \phi)}=\frac{\partial \mathbf{x}}{\partial \rho}\times\frac{\partial \mathbf{x}}{\partial \Theta}\cdot\frac{\partial \mathbf{x}}{\partial \phi}. \tag{6}$$

From Eq. (4), we see that the Jacobian for flux coordinates with Ψ as the independent variable is determined by the B^{ϕ} component of the magnetic fields,

$$\mathscr{J}^{\Psi}=\frac{\partial(x, y, z)}{\partial(\Psi, \Theta, \phi)}=\frac{1}{\nabla\Psi\times\nabla\Theta\cdot\nabla\phi}=\frac{1}{B\cdot\nabla\phi}.$$

The ratio of the two Jacobians is $\Psi'(\rho)$. In practice, the numerical evaluation of the two Jacobians gives slightly different angular dependencies due to numerical errors. We take

$$d\Psi/d\rho=\langle\mathscr{J}^{\rho}(1/\mathscr{J}^{\Psi})\rangle=\langle\mathscr{J}^{\rho}B^{\phi}\rangle, \tag{7}$$

where the brackets $\langle\ldots\rangle$ denote an average over poloidal and toroidal angles.

Equation (7) determines the toroidal flux inside each flux surface. The poloidal flux is given in terms of the toroidal flux by the rotational transform. Our algorithm for calculating the rotational transform will be described in Section III.

We represent all functions in magnetic coordinates as a two-dimensional Fourier sum on each of $L+1$ magnetic surfaces. The surfaces are labeled by the discrete index l, where $0 \leqslant l \leqslant L$. The surface $l=0$ corresponds to the magnetic axis, while $l=L$ corresponds to the outer boundary of the toroidal domain. Each Fourier expansion has the form

$$\sum_{n,\,m} a^l_{n,\,m} \begin{pmatrix} \sin \\ \cos \end{pmatrix} (nN\phi - m\Theta), \tag{8}$$

where N is the number of periods, and where the indices n, m obey the inequalities:

$$|n| \leqslant \mathcal{N}; \qquad 0 \leqslant m \leqslant \mathcal{M}. \tag{9}$$

The values of L, $\mathcal{M}$, and $\mathcal{N}$ are specified as input parameters to the code. The asymmetric bounds in Eq. (9) are motivated by the symmetry properties of the Fourier coefficients $a_{n,\,m}$,

$$a^l_{n,\,m} = \pm a^l_{-n,\,-m}.$$

This follows from the even/odd symmetry of the functions with respect to double reflection in Θ and in ϕ. The quantities x, B^Θ, and B^ϕ (and hence $\mathcal{J}^\Psi = 1/B^\phi$) have even symmetry while y, z, B^ρ, and h have odd symmetry. In this representation, one solution to Eq. (5) is

$$f(k\,\Delta\rho, \Theta, \phi) = \sum_{n,\,m} \frac{(\mathcal{J}^\Psi h)^l_{n,\,m}}{(nN - \iota m)} \cos(nN\phi - m\Theta). \tag{10}$$

The general solution consists of this solution plus any function of ρ.

The representation of Eq. (8) needs $(L+1)(\mathcal{M}+1)(2\mathcal{N}+1)$ modes stored, roughly half the coefficients that would be needed if we did not take advantage of the symmetry properties. The redundant elements $a^l_{n,\,0}$ and $a^l_{-n,\,0}$, $n>0$, are retained to simplify the code. For nonaxisymmetric fields ($\mathcal{N}>0$), the presence of these redundant terms provides a convenient check on the symmetry of our results.

As illustrations, we now discuss the solution of Eqs. (2) and (3). Equation (2) is a homogeneous magnetic differential equation. The solution of such an equation is constant on each magnetic surface. The value of the pressure on each surface is an integration constant which is given, typically as a function of the flux. Equation (7), together with the equations defining the transformation to magnetic coordinates, then determines the pressure everywhere in the region of interest.

Equation (3) follows [2] from the MHD equilibrium equations:

$$\mathbf{J} \times \mathbf{B} = \nabla p, \tag{11}$$

$$\nabla \times \mathbf{B} = \mathbf{J}. \tag{12}$$

The current is divergence free and satisfies $\mathbf{J}\cdot\nabla\Psi=0$. It follows that $\mathbf{J}$ can be written in magnetic coordinates in the form

$$\mathbf{J}=\left(I'(\Psi)-\frac{\partial v}{\partial\Theta}\right)\nabla\Psi\times\nabla\Theta+\left(\frac{\partial v}{\partial\phi}-g'(\Psi)\right)\nabla\phi\times\nabla\Psi. \tag{13}$$

Here v is a single-valued function to be determined, while $I'(\Psi)=dI/d\Psi$ and $g'(\Psi)=dg/d\Psi$ are the profiles of the net toroidal and poloidal current, respectively. Typically, either $I'(\Psi)$ or $g'(\Psi)$ is given, and the other can then be determined from the equilibrium equations. For our purposes here, we may regard both these profiles as given. Substituting Eq. (13) into Eq. (11) gives Eq. (3). Following the prescription given at the beginning of this section, we obtain the solution,

$$v=\frac{dp}{d\Psi}\sum_{n,m}\frac{\mathscr{J}^{\Psi}_{n,m}}{(nN-\iota\!\!\!/ m)}\sin(nN\phi-m\Theta), \tag{14}$$

which we have expressed in terms of the Fourier components of the Ψ Jacobian,

$$\mathscr{J}^{\Psi}(\Psi,\Theta,\phi)=\sum_{n,m}\mathscr{J}^{\Psi}_{n,m}\cos(nN\phi-m\Theta). \tag{15}$$

We will discuss an explicit solution for the current in the ATF stellarator in Section V.

III. Fourier Decomposition in Magnetic Coordinates

As we have discussed, the crucial step in our code is the determination of the two-dimensional Fourier transform in magnetic coordinates from the one-dimensional Fourier decomposition along a field line. In this section we show how a precise determination of the rotational transform allows us to calculate easily the required two-dimensional Fourier transform. We also describe our algorithm for efficiently evaluating the rotational transform.

We begin by observing that Eq. (4) can be written in the form

$$\mathbf{B}=\nabla\Psi\times\nabla(\Theta-\iota\!\!\!/\phi),$$

where we have now specialized to our choice of the uniform toroidal angle ϕ as the magnetic coordinate Φ. This equation implies that a field line is defined by the intersection of two surfaces, a constant Ψ surface and a constant $\Theta-\iota\!\!\!/\phi$ surface. In these coordinates $\Theta-\iota\!\!\!/\phi$ is constant along the field lines. (This is equivalent to saying that the field lines are straight.) That is, we can write $\Theta=\iota\!\!\!/\phi$ on any given field line, since the constant can be set to zero by shifting a coordinate origin.

It follows that there is a simple relation between the one-dimensional Fourier transform along a field line of a function a and its two-dimensional Fourier transform in magnetic coordinates. Since $a(\Psi,\Theta,\phi)$ is periodic in the two angles, with a periodicity length of $2\pi/N$ in the toroidal direction and 2π in the poloidal direction, it can be represented by a Fourier series on each flux surface:

$$a(\Psi, \Theta, \phi) = \sum_{n,m} a_{n,m}(\Psi) \begin{pmatrix} \sin \\ \cos \end{pmatrix} (nN\phi - m\Theta). \tag{16}$$

As discussed in the previous section, we have assumed that a has odd (sine) or even (cosine) symmetry with respect to the double reflection $\Theta \to -\Theta$ and $\phi \to -\phi$ (the generalization to the nonsymmetric case is straightforward). If we evaluate a along a field line (which lies in a constant Ψ surface), Θ and ϕ are linearly related as described above. Equation (16) then assumes the form:

$$\sum_{n,m} a_{n,m}(\Psi) \begin{pmatrix} \sin \\ \cos \end{pmatrix} [(nN - \iota\!\!\!-m)\,\phi]. \tag{17}$$

The Fourier spectrum of the function a evaluated along a field line therefore consists of a set of discrete lines. The frequencies of these lines can be used to determine $\iota\!\!\!-$. The amplitudes of these lines determine the two-dimensional Fourier coefficients in magnetic coordinates, $a_{n,m}$ in Eq. (16). Given **B** at all points in a toroidal domain, it is easy to follow field lines numerically to high accuracy. A fast Fourier transform (FFT) can then be used to obtain the spectrum along the field lines, and thus the Fourier representation in magnetic coordinates for the corresponding magnetic surfaces.

In practice, given only the one-dimensional Fourier representation of a function along a field line, the identification of the spectral lines with those of the two-dimensional representation on the magnetic surface containing the field line is not straightforward. This identification becomes complex and unreliable when the smallest frequency difference between the spectral lines is less than the minimum of $\iota\!\!\!-$ and 1. Further, one must take into account the uncertainty due to numerical error in the frequencies of the spectral lines, which is not known *a priori*.

Our solution is first to determine $\iota\!\!\!-(\Psi)$ to high accuracy. The mode numbers m and n associated with each of the spectral lines in the one-dimensional spectrum, Eq. (17), can then be identified uniquely. For this purpose, an efficient method for calculating $\iota\!\!\!-$ is needed. A conventional approach uses the definition of the transform as a winding number and determines numerically $\lim(\theta/\phi)$ as $\phi \to \infty$, where θ is any convenient coordinate which increases by 2π in going around the magnetic axis, and which is single valued in the toroidal direction. This method for evaluating $\iota\!\!\!-$ can be costly since it converges linearly with the length of the field line and many orbits around the torus may be needed. The computer time consumed by the code for this method increases somewhat faster than linearly with the distance over which the field lines must be followed, because the local error in the numerical integration must be decreased as the integration length is increased to give a fixed global integration error.

To improve this procedure for calculating $\iota\!\!\!-$, we observe that $\theta - \iota\!\!\!-\phi$ is a periodic function and can be written as a Fourier series of the form of Eq. (16). This expression can be used to improve systematically an estimate for ι given by $\lim(\theta/\phi)$. The improved estimates of ι converge towards the correct value exponentially with the field line length. We therefore determine ι on each flux surface by an iterative procedure that chooses ι at each step to minimize the expression

$$F(\iota)=\sum_{j}\left\{\theta(\phi_j)-\iota\phi_j-\sum_{n,m}\theta_{n,m}\sin[(nN-\iota m)\,\phi_j]\right\}^2. \tag{18}$$

Here the outside sum goes over equally spaced sample points, $\phi_j=j\,\Delta\phi$, along the field line. The values $\theta(\phi_j)$ have been determined by following a field line. The inner sum goes over the discrete range, $-\mathcal{N}\leqslant n\leqslant\mathcal{N}$, $0\leqslant m\leqslant\mathcal{M}$. For the first step we take all $\theta_{n,m}=0$. At each subsequent step, we use the current value of ι, together with an FFT of the quantity $\theta-\iota\phi$, to determine the $\theta_{n,m}$. Typically two or three iterations suffice. The convergence criterion is discussed at the end of Section IV.

For the minimization of F, we use a NAG library routine [12] (e04bbf) which makes use of derivative information. From Eq. (18) it is straightforward to obtain an expression for the derivative of F with respect to ι. In following field lines, the rate at which we sample points is determined by the shortest wavelength we can resolve in our finite Fourier representation,

$$\Delta\phi\leqslant\frac{\pi}{N\mathcal{N}+|\iota|_{\max}\mathcal{M}}, \tag{19}$$

where $N\mathcal{N}$ and $\mathcal{M}$ are the maximum toroidal and poloidal mode numbers, and $|\iota|_{\max}$ is the maximum value of $|\iota|$. The code requires an upper bound on the rotational transform to be specified as an input parameter to determine the sampling rate.

Rapid convergence of our algorithm to calculate ι occurs to the extent that the dominant dependence of F on ι comes from the $\iota\phi$ term. This can be guaranteed by following the field line sufficiently far. In practice, the distance for which the field lines are followed is determined by the accuracy requirement of the one-dimensional FFT along the field lines, as discussed in Section IV. This distance is sufficient to guarantee an accurate initial guess for the algorithm. The algorithm is efficient in making optimal use of the information obtained by following a field line over a given distance.

We illustrate the convergence and accuracy of the algorithm for a simple model field. We consider an analytical field of the form of Eq. (4), with

$$\Psi=r^2, \qquad \Theta=\theta+\tfrac{1}{2}\,\theta_2\sin 2\theta,$$

where θ is the uniform geometric poloidal angle. In this expression, ι and θ_2 are constants to be specified. The field is divergence free, but of course not necessarily an equilibrium field. For our example we take $\iota=0.14$ and $\theta_2=0.2$. We will see in Section V that this θ perturbation is considerably larger than that for a typical stellarator vacuum field. We run the code with 10 poloidal modes, following the field lines for nine toroidal circuits of the torus. For this case, we find that the accuracy with which the code determines ι is worst on the outermost flux surface ($r=1$). On this surface, the first iteration of the algorithm gives $\iota=0.13968479$, the second gives $\iota=0.13999842$, and the third iteration gives $\iota=0.13999990$. Subsequent iterations give only a slight further improvement. The error in the final result is attributable to the finite number of poloidal modes retained.

We now explain how the various quantities x, y, z, h, $\mathscr{J}^{\Psi}$, and θ are evaluated along the field lines of $\mathbf{B}$. The field lines are followed in a user-specified toroidal coordinate system (r, ϑ, ϕ), where ϕ is constrained to be the uniform toroidal angle. The angle ϑ in this coordinate system differs, in general, from both the magnetic coordinate Θ and the uniform geometric angle θ. In particular, the origin of the (r, ϑ, ϕ) coordinate system does not generally coincide with the magnetic axis, while θ measures an angle about the magnetic axis. The (r, ϑ, ϕ) coordinates can be specified as input to the code by giving the Fourier transform of the Cartesian vector $\mathbf{x}$ in ϑ and ϕ for a set of equally spaced surfaces in r. As a default, the code takes ϑ and ϕ to be uniform poloidal and toroidal angles, and takes r to be the distance from a coordinate axis whose location is indepenent of ϕ. The location of this axis is determined by specifying the aspect ratio of the toroidal region. The option to define a different coordinate system allows for efficient representation in domains whose boundaries are far from circular.

The contravariant components of the magnetic field, $(B^r, B^\vartheta, B^\phi)=(\mathbf{B}\cdot\nabla r, \mathbf{B}\cdot\nabla\vartheta, \mathbf{B}\cdot\nabla\phi)$, are specified as input to the code. These components are specified either by giving their values on a three-dimensional grid, with uniform grid spacing in r, ϑ, and ϕ, or by giving their Fourier coefficients on a set of radial coordinate surfaces uniformly separated in r. If the values are given on a grid, the code uses a fast Hartley transform (see Section IV) to Fourier decompose on the radial coordinate surfaces. As described below, the code interpolates the Fourier coefficients radially to follow magnetic field lines. The magnetic field can also be specified by an analytic expression. This mode of operation does not require any interpolation to follow field lines and is therefore useful for rapid study of a range of parameters.

To avoid problems at the coordinate axis due to the singular behavior of B^ϑ (which goes as $1/r$ for r near zero), field lines are followed in the "pseudo-Cartesian" coordinates

$$\xi = r\cos\vartheta, \qquad \eta = r\sin\vartheta. \tag{20}$$

The ordinary differential equations to be integrated are then:

$$\frac{d\xi}{d\phi}=\frac{B^r}{B^\phi}\cos\vartheta-\frac{B^\vartheta}{B^\phi}\,r\sin\vartheta,$$

$$\frac{d\eta}{d\phi}=\frac{B^r}{B^\phi}\sin\vartheta+\frac{B^\vartheta}{B^\phi}\,r\cos\vartheta.$$

For an initial condition (ξ_0, η_0, ϕ_0), we integrate these equations with respect to ϕ and store the coordinates at equally spaced intervals $\Delta\phi$, giving a sequence of vectors $(\xi_j, \eta_j, j\,\Delta\phi)$ at which we evaluate the functions to be sampled. We use a variable order, variable stepsize predictor-corrector algorithm (the NAG routines d02caf and d02cbf), which evaluates values at intermediate sample points by interpolation. The code assumes (and verifies) that $B^\phi > 0$ at each step (this simply states that the Jacobian between magnetic and Cartesian coordinates is well defined in the toroidal domain). This condition is violated for reversed-field pinches, but is almost always true for tokamaks and stellarators. The contraint on the toroidal component of **B** is not fundamental to our method, but comes from our choice of the toroidal angle.

The contravariant components of **B**, as well as the quantities x, y, z, and h, in general must be evaluated along the field lines by interpolation. We interpolate the Fourier coefficients of these quantities as a function of r, and then sum the interpolated Fourier coefficients to obtain the value of the function at a given ϑ and ϕ. This procedure allows us to represent accurately the high frequency Fourier components near the origin by using the analyticity of these functions. Analyticity implies that the functions have a convergent Taylor series expansion around the

components using a low order interpolotation scheme would give poor accuracy near the origin. We factor out the leading r^m power of the Fourier modes, for all m up to a maximum value $m = \text{maxexp}$, and then use a cubic spline to interpolate the normalized Fourier coefficients in the radial direction. For $m > \text{maxexp}$ we factor out only r^{maxexp}. The *maxexp* cutoff is necessary because the numerically evaluated Fourier coefficients are accurate only to a finite precision. When m is sufficiently large that the Fourier coefficients become smaller than this precision at the first few grid points near the origin, pulling out an r^m amplifies the numerical errors there. In practice, we find that setting $\text{maxexp} \approx 3$ gives satisfactory results.

For our radial spline interpolation we use "not-a-knot" boundary conditions [13] at $r = 0$ and $r = 1$. We have found that the so-called "natural" boundary condition available in many spline packages, which imposes the constraint that the second derivative of the function vanish at the boundaries, gives a serious deterioration in accuracy near the boundaries. This is not too surprising at the outer boundary, since there is no reason to expect the Fourier coefficients to satisfy this constraint there. At the origin, however, the solution should satisfy that boundary condition if we pull an r^{m-1} out. The problem again arises at large m because of the finite precision of the numerical Fourier transform.

We need to find the magnetic axis to define the coordinate θ, and to choose initial conditions $(r_0, \vartheta_0, \phi_0)$ for our field line integrations on a sequence of magnetic surfaces radiating out from the axis. The sequence of initial conditions implicitly defines the radial magnetic coordinates, ρ_j. The intersection of the magnetic axis with a vertical symmetry plane of the domain must lie in the midplane of the torus. We solve for this point by labeling the intersection of the two planes (the $\phi = 0$ plane and the midplane) with the Cartesian coordinate x, and by defining a function $f(x)$ to be the distance of the point defined by the return map from the initial point x. Any magnetic axis is a zero of f. We search for this zero by using a secant method (NAG routine c05ajf [12]). We choose our tolerance for this routine to be an order of magnitude smaller than the tolerance in the Fourier decomposition.

In restricting our search for the magnetic axis to one dimension, we have made essential use of the assumed symmetry. In the absence of symmetry, it would be necessary to search a two-dimensional plane. The magnetic axis would again correspond to a zero of the distance defined by the return map from the initial point.

Given the location of the magnetic axis, the code follows $L+1$ field lines, with initial points lying on the intersection of a symmetry plane with the midplane, equally spaced in the Cartesian x coordinate between the magnetic axis and the boundary (including the axis and the boundary). The $L+1$ field lines determine $L+1$ coordinate surfaces, equally spaced in ρ between the magnetic axis ($\rho = 0$) and the outermost surface ($\rho = 1$), thus defining our ρ coordinate. The code simultaneously follows L of the field lines (excluding the surface) by treating the L pairs of associated ode's as $2L$ "coupled" equations. This allows the expensive field evaluations to be vectorized. We pay the price that the step size must be the same on all the field lines.

Given the values of a function at the sample points $(\xi_j, \eta_j, j\,\Delta\phi)$, we can perform a discrete Fourier transform in ϕ. This is complicated by the fact that the data is not periodic in ϕ. In the following section, we describe how we perform Fourier decompositions along the magnetic field lines, and we discuss the associated errors.

IV. Fourier Transform along Field Lines and Its Accuracy

In the previous section we showed how a precise determination of ι allows us to identify directly the spectral lines of the one-dimensional Fourier transform along a field line with the spectral lines of a two-dimensional Fourier transform in magnetic coordinates. We then presented an algorithm for calculating ι efficiently and precisely. At each step, this algorithm requires a Fourier decomposition of $\theta - \iota\phi$ along field lines, using the current value of ι. Similarly, the Fourier decomposition of any function in magnetic coordinates requires, as a first step, the Fourier decomposition of that function along field lines. In this section we describe how we Fourier decompose functions along magnetic field lines, and we analyze the resulting numerical errors. We will see that the accuracy with which we can determine the Fourier coefficients numerically is limited by the distance over which the field lines have been followed. Following field lines is the most time-consuming part of the code. We therefore want to follow the field lines the minimum distance consistent with the required precision. As we will see, the required distance can be strongly influenced by the presence of low order rational surfaces.

We need to obtain one-dimensional Fourier decompositions of the form given by Eq. (17). This expression is not periodic in ϕ (except for the special cases where ι is rational). To complicate matters further, near those rational surfaces $\iota/N = n/m$ where a_{nm} is nonzero, the periodicity length of the $\sin[(nN - \iota m)\phi]$ term becomes large, making this term difficult to distinguish from the $m = 0$, $n = 0$ term. The problem arises more generally whenever the periodicities of any two different terms in Eq. (17) are nearly equal. These issues are the primary subject of this section. Before turning to them, we say a few words about the numerical method we use for performing fast Fourier transforms. At the end of this section we will turn to the question of interpolating the amplitudes in Fourier space, to go from a discrete Fourier transform to the required coefficients at $nN - \iota m$.

To perform our fast Fourier transforms, we use a fast, highly vectorized, machine language routine written by O. Buneman [14]. This routine uses the Hartley kernel, cos + sin, rather than the $\cos + i \sin$ of the usual Fourier transform. This avoids complex arithmetic when transforming real data. For odd functions [such as $\theta(\phi)$], the coefficients of the sin terms, s_j, are expressed in terms of the fast Hartley transform (FHT) coefficients, H_j, by

$$s_j = (H_j - H_{n-j})/\sqrt{n},$$

where n is the number of points. For even functions, the corresponding formulas for the coefficients of the cosine terms are:

$$\begin{aligned} c_0 &= H_0/\sqrt{n}, \\ c_{n/2} &= H_{n/2}/\sqrt{n}, \\ c_j &= (H_j + H_{n-j})/\sqrt{n}, \qquad 0 < j < n/2. \end{aligned}$$

In determining the FHT coefficients for symmetric functions, the known symmetry property could, in principle, be used to reduce the required integration length by a factor of two. That is, we could integrate from $\phi = 0$ to some $\phi = \phi_f$, rather than from $-\phi_f$ to ϕ_f. We found that this integration over half the interval introduces considerable numerical noise in the tail of the Fourier spectrum. (Note that although we apply our FHT from $-\phi_f$ to ϕ_f, we only follow the field line from $\phi = 0$ to $\phi = \phi_f$, and we reflect the data using the appropriate symmetry.)

The functions whose Fourier representation we seek are not, in general, periodic. We therefore must multiply them by an appropriate window function to reduce the discontinuity at the boundary of the periodic extension before performing a numerical Fourier decomposition. We use a Gaussian window, proportional to $\exp[-(\phi/\phi_g)^2]$. As measured by various figures of merit, the Gaussian window does nearly as well as the considerably more complicated windows which are also in common use [15]. It has the advantage that analytical estimates of expected errors can be obtained. An outline of the derivation of the analytical estimates is given in Appendix A. The relative error in each Fourier coefficient due to the finite integration length ϕ_f is

$$\frac{1}{\sqrt{\pi}} \frac{\phi_g}{\phi_f} \exp[-(\phi_f/\phi_g)^2]. \tag{21}$$

The error in the FHT due to interference by a second mode of amplitude A is

$$A\left[\exp\left(-\frac{1}{4}(\Delta k\phi_{\mathrm{g}})^2\right)-\frac{1}{\sqrt{\pi}}\frac{\cos(\Delta k\phi_{\mathrm{f}})\exp[-(\phi_{\mathrm{f}}/\phi_{\mathrm{g}})^2](\phi_{\mathrm{f}}/\phi_{\mathrm{g}})}{(\phi_{\mathrm{f}}/\phi_{\mathrm{g}})^2+(1/4)(\Delta k\phi_{\mathrm{g}})^2}\right], \tag{22}$$

where

$$\Delta k=(n_1N-\iota m_1)-(n_2N-\iota m_2) \tag{23}$$

is the distance in k space between the modes, with (n_1N, m_1) and (n_2N, m_2) the mode numbers of the two modes. This interference, often called "spectral leakage," arises from the nonorthogonality of the Fourier modes in Eq. (17). The spectral leakage from a set of modes is linearly additive. Equations (21) and (22) have been obtained by expressing the integral of a truncated Gaussian times a trigonometric function in terms of the error function, and using an asymptotic expansion of the error function in $\phi_{\mathrm{f}}/\phi_{\mathrm{g}}$ [16]. The second term in Eq. (22) dominates in the large k tail of the Fourier spectrum and determines the smallest amplitudes that can be correctly identified. It gives the tail of the spectrum a noisy appearance when the spectral amplitudes are plotted logarithmically.

As we get close to a low order rational surface, Δk can become small. This requires an increased integration length, ϕ_{f}, to maintain the accuracy of the numerical Fourier transform. The problem only arises near those surfaces where $\iota/N=n/m$, with $0\leqslant m\leqslant 2\mathscr{M}$ and $-2\mathscr{N}\leqslant n\leqslant 2\mathscr{N}$. Right at such a rational surface Δk can become zero, making it impossible to apply our Fourier transform method there. To handle this problem, we determine the nearly overlapping (in k-space) Fourier coefficients on the coordinate surface closest to such a rational surface by interpolation. Taking the average of the coefficients with the same m and n on the two neighboring coordinate surfaces gives a second-order accurate interpolation because of our uniform radial grid. Before averaging, we pull out an r^m factor, up to a maximum value of $m=\text{maxexp}$, as described in the discussion of interpolation in Section III. The code chooses ϕ_{g} and ϕ_{f} to make the errors everywhere else—as described by Eq. (21) and the first term in Eq. (22)—less than ftprec/2 and A ftprec/2, respectively. The precision of the Fourier amplitudes, *ftprec*, is an input parameter to the code. Note that the second term of Eq. (22) is bounded by Eq. (21). In the presence of low order rational surfaces, the Δk which determines the required integration length is determined by the local shear and the radial grid spacing. As we increase the number of radial grid points, the minimum Δk decreases, requiring increased ϕ_{f}. For a fixed number of radial grid points, decreasing the shear also increases the required integration length.

The constraint on the first term of Eq. (22) gives

$$\phi_{\mathrm{g}}=\frac{\sqrt{-4\ln(\text{ftprec}/2)}}{\Delta k}. \tag{24}$$

The code solves for ϕ_{f} from Eq. (24) and the constraint on Eq. (21), again using a secant method (NAG routine c05ajf). In determining ϕ_g and ϕ_f, rather than let Δk be the distance between modes, the code uses an effective (somewhat smaller) Δk to be described below.

We take the interval over which we carry out the FHT, $[-\phi_{\text{fft}}, \phi_{\text{fft}}]$, to be greater than the interval over which we follow field lines, $[-\phi_f, \phi_f]$. This effectively interpolates the spectral amplitudes in k. We fill the interval between ϕ_f and ϕ_{fft} by extending the tail of the Gaussian. The length of the tail is chosen so that there are at least five points between the spectral peaks,

$$\phi_{\text{fft}} \geqslant \frac{10\pi}{\Delta k_{\min}}, \tag{25}$$

where $\Delta k_{\min}$ is the minimum value of Δk. This gives a constraint on the number of points needed in the discrete Fourier transform,

$$N_{\text{fft}} \geqslant \frac{10}{\Delta k_{\min}} (N_{\mathcal{N}} \mathcal{N} + |\iota|_{\max} \mathcal{M}). \tag{26}$$

In addition, N_{fft} must be a power of 2.

We must interpolate between points in k space to evaluate the transform at $nN - \iota m$. (This gives the Fourier coefficient a_{nm}.) To do this, we fit the three points nearest $k = nN - \iota m$ to a Gaussian

$$f_g \exp[-\alpha(k - k_0)^2].$$

If f_1, f_2, f_3 are the values of the FFT at the points k_1, k_2, k_3, then:

$$k_0 = \frac{1}{2} \frac{\ln(f_2/f_1)(k_2^2 - k_3^2) - \ln(f_3/f_2)(k_1^2 - k_2^2)}{\ln(f_3/f_2)(k_2 - k_1) - \ln(f_2/f_1)(k_3 - k_2)}, \tag{27}$$

$$\alpha = \frac{(k_1 - k_0)^2 - (k_2 - k_0)^2}{\ln(f_2/f_1)}, \tag{28}$$

$$f_g = f_1 \exp\left(\frac{(k_1 - k_0)^2}{\alpha}\right). \tag{29}$$

Rather than evaluating the Gaussian at $k = nN - \iota m$, we use f_g for the desired amplitude. This accelerates the convergence of our iterative scheme for ι. Although this scheme is extremely accurate if we are near the peak of the Gaussian, it is poorly behaved if we are on the tail of the Gaussian, or if we are so far from the peak that the behavior is nongaussian. For that reason, we first perform some checks, before applying Eqs. (27)–(29), to be sure that we are near the peak. If any of these tests are failed, we use an alternative value for the amplitude. If the three FHT values are not all the same sign we know that we are far from the peak of a Gaussian, so we take the amplitude to be zero. Otherwise, if the center point does not have the largest amplitude we use the value of f_1 or f_3, whichever is larger.

For our interpolation, we require the FFT coefficients at three points near $k = nN - \iota m$. To ensure that the amplitudes at each of these points is within a precision *ftprec*, we take the Δk in Eq. (22) to be the minimum distance in k space between any of these points and a neighboring spectral peak.

How accurately do we need to determine ι? For $k = nN - \iota m$, an error $\Delta\iota$ in ι gives an error $\delta k = -m\,\Delta\iota$ in k. We want this to be less than half the distance between the points in our spectrum,

$$\delta k = -m\,\Delta\iota < \pi/\phi_{\mathrm{fft}}.$$

Once this is satisfied, our interpolation algorithm is eliminated as a source of numerical error. There is an additional error which arises directly from the Fourier decomposition of $\theta - \iota\phi$, due to the inaccuracy in the numerical calculation of ι. Relative to spectral leakage, this contributes an error $-4\ln(\mathrm{ftprec}/2)\ \Delta\iota/\Delta k$. The code verifies that both these contributions to the numerical inaccuracy of the Fourier decomposition are small.

V. Example and Discussion

In the preceding sections we have described our numerical algorithm and its implementation. In this section we present, as a concrete example, the numerical solution of Eq. (3) for a stellarator vacuum field. This gives the lowest order (in β) pressure-driven current. We consider an $l = 2$ stellarator with parameters corresponding to the ATF device at Oak Ridge [17].

To obtain our vacuum field we solve Laplace's equation for the scalar potential of the magnetic field,

$$\mathbf{B} = \nabla\chi, \tag{30}$$

with the boundary condition $\mathbf{n}\cdot\nabla\chi = 0$, where $\mathbf{n}$ is the unit normal to the outer flux surface. Laplace's equation is solved using a Fourier representation in the toroidal and poloidal angles, and second-order finite differences in the radial direction, with the resulting block tridiagonal matrix inverted by Gaussian elimination [3, 18]. For the example discussed in this section we used twenty radial grid surfaces ($L = 20$) and 77 Fourier modes ($0 \leqslant m \leqslant 10$, $-3 \leqslant n \leqslant 3$). The outer flux surface is specified by the Bessel function solution for the cylindrical vacuum field,

$$\chi = B_0\phi + b_2 I_2(Nr/R)\sin(2\vartheta - 2N\phi), \tag{31}$$

where $N = 12$ is the number of periods, and $R = 7$ is the major radius. The ratio b_2/B_0 determines the rotational transform. We choose a value which is appropriate for ATF, $2Nb_2/B_0 = 0.633$, giving a rotational transform which ranges from 0.308 at the axis to 0.975 at the edge.

The vacuum field has rational surfaces for $12.3 < m/n < 39.0$. The lowest order rational surface is at $\iota = \frac{12}{13}$ ($n = 1$, $m = 13$). For $\mathcal{M} \leqslant 6$ there is insufficient angular resolution for the code to see the rational surfaces, and their presence does not pose any difficulties. When $\mathcal{M} = 7$, $\mathcal{N} \geqslant 1$, the $n = 1$, $m = 7$ mode is indistinguishable from the $n = 0$, $m = 6$ mode at the $\iota = \frac{12}{13}$ rational surface. Similarly, the $n = 0$, $m = 7$ mode is indistinguishable from the $n = 1$, $m = 6$ mode there. There is a single pair of indistinguishable modes at the $\iota = \frac{6}{7}$ surface. If there is a coordinate surface very near the $\iota = \frac{12}{13}$ or $\iota = \frac{6}{7}$ rational surface, $\Delta k_{\min}$ is small. (Throughout this section we use $\Delta k_{\min}$ to denote the minimum value of Δk on a given coordinate surface. The quantity Δk was defined in Section IV.) As discussed in the previous section, this problem is handled by using interpolation to calculate the amplitudes of the nearly indistinguishable modes on the coordinate surface nearest the rational surface. As $\mathcal{M}$ is increased, the number of required interpolations also increases.

In practice, it does not always happen that a coordinate surface falls close enough to the rational surface to pose a problem. The criterion that we have adopted is to interpolate on those interior coordinate surfaces where $\Delta k_{\min}$ has a local minimum as a function of the radial coordinate ρ, if the local value of $\Delta k_{\min}$ is smaller than that at the first and last radial coordinate surfaces. We then calculate the global minimum of Δk, disregarding these near rational coordinate surfaces on which we will interpolate. This is the value of Δk that we use to determine the required integration lengths. Even on the near rational coordinate surfaces, we interpolate only those modes belonging to mode pairs with Δk less than this global minimum. For a monotonic ι profile, $\Delta k_{\min}$ is typically a monotonic function of ρ as long as $\mathcal{M}$ is sufficiently small that the code does not see any rational surfaces. Even for larger $\mathcal{M}$, $\Delta k_{\min}$ may remain monotonic if the coordinate surfaces do not happen to fall near any of the low order rational surfaces. In that case no interpolation is required, and the code does not interpolate. For nonmonotonic ι, $\Delta k_{\min}$ is generally nonmonotonic even in the absence of low order rational surfaces. The code does interpolate in that case whenever the required integration length can be decreased.

For our ATF example we take $\mathcal{M} = 10$, $\mathcal{N} = 3$. The code interpolates Fourier coefficients on surfaces 15, 17, and 19, corresponding to $\Delta k_{\min} = 0.024$, 0.097, 0.173, respectively, with $\iota = 0.598$, 0.713, 0.871. Calculating the global minimum of Δk over all the surfaces, excluding those on which we interpolate, gives a value of 0.190. To obtain an accuracy ftprec $= 10^{-5}$ the code follows the field lines 20 times around the torus. The largest amplitude mode interpolated in calculating the Fourier transform of $\theta - \iota\phi$ is the $m = 4$, $n = 1$ mode on surface 19, with an interpolated amplitude of 9.3×10^{-5}. Its values on the neighboring coordinate surfaces are 9.0×10^{-5} and 9.6×10^{-5}, with a difference of less than *ftprec* between them. The largest Fourier mode is the $m = 2$, $n = 1$ component, with a maximum amplitude of about 0.06 at $r = 0.6$. This is considerably smaller than the θ perturbation considered in the example of Section III.

The Fourier components of the pressure-driven current are determined directly in terms of those of the Jacobian by Eq. (14). Note that although there may be near-zero terms in the denominator of this equation, the Jacobian is $1/B^{\varphi}$, a finite and well-behaved quantity. This enables us to calculate accurately near-resonant terms. The solution requires only that we Fourier decompose $1/B^{\phi}$. In doing so, we again interpolate the nearly indistinguishable mode pairs. We have set maxexp $= 5$ for the interpolations. The interpolated mode having largest amplitude is now the $m=4$, $n=2$ mode on surface 19, with an interpolated amplitude of 0.02902. At the neighboring coordinate surfaces, the mode has amplitudes of 0.02236 and 0.03718. The error in the interpolation is equal to the second derivative of the mode amplitude with respect to ρ times $\Delta\rho^2/2$, where $\Delta\rho = 0.05$ is the radial grid spacing. Evaluating the second derivative numerically, we estimate the error to be $6. \times 10^{-5}$. For this particular mode the numerical accuracy is therefore dominated by the accuracy of the radial interpolation. We will see that the overall accuracy for this calculation is determined by the amplitude of the neglected Fourier modes.

Figure 1 shows the numerically calculated Fourier transform of $1/B^{\phi}$ with respect to ϕ along a magnetic field line lying in the outermost flux surface. The horizontal coordinate is the mode number, with peaks at $nN - \text{ɫ}m$. Instead of discrete spectral lines, the numerical fast Fourier transform with a Gaussian window gives a set of Gaussian shaped curves. They appear as narrow quadratic curves on our logarithmic plot, centered at the points $nN - \text{ɫ}m$. The width of the Gaussians is about $3.3/\phi_g = 0.09$. The four highest peaks have been labeled in Fig. 1 with their mode numbers, (n, m). The clustering of peaks in this figure is due to the toroidal coupling to the helical modes satisfying $m = 2n$. For the fast Fourier transform, we have evaluated 4096 Fourier coefficients along each field line, giving a horizontal

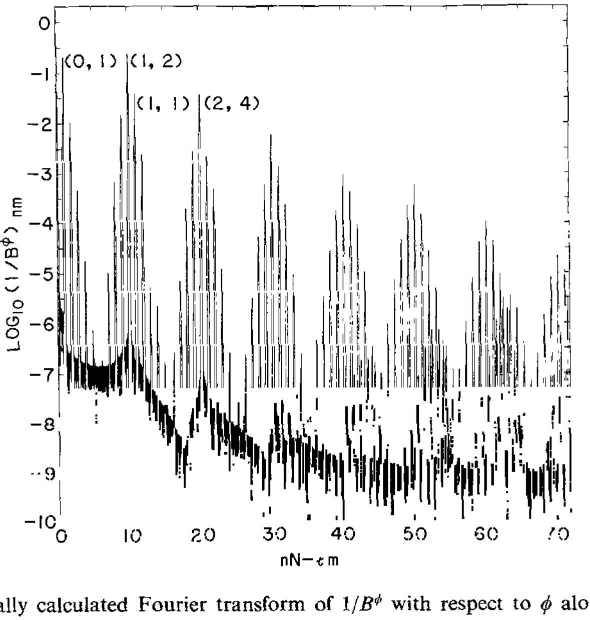

FIG. 1. Numerically calculated Fourier transform of $1/B^{\phi}$ with respect to ϕ along a magnetic field line lying in the outermost flux surface for the ATF vacuum field. The four largest mode amplitudes have been labeled with their mode number, (n, m).

resolution of about 0.018. The rotational transform on this flux surface is calculated to be $\iota = 0.9752$, with an associated value of $\Delta k_{\min} = 0.2709$. It is apparent from Fig. 1 that the largest neglected Fourier mode has an amplitude nearly equal to $1. \times 10^{-3}$ at the outer flux surface. It has mode numbers $m = 8$, $n = 4$. The corresponding Fourier mode of v has an $nN - \iota m$ in the denominator, giving a (neglected) contribution of roughly $2. \times 10^{-5}$ in amplitude.

Figure 2 shows the ρ dependence of the four largest Fourier modes. The mode amplitudes are largest at the outer boundary. They are smooth functions of ρ, going like ρ^m near the coordinate origin, but deviating from this near the outside.

In solving the magnetic differential equation for the lowest order pressure-driven current in a stellarator vacuum field, we have encountered a number of subtle issues related to the presence of low order rational surfaces. We have seen how these issues are handled automatically by our code, with no need for intervention. Integration lengths are adjusted automatically to compensate for proximity to the low order rational surfaces. Nearly indistinguishable mode pairs are calculated by interpolation. Because the code works in magnetic coordinates, the small $nN - \iota m$ denominators factor out, and there is no need to manipulate nearly singular quantities. In the absence of large islands and stochastic regions, the algorithm presented here provides an accurate, efficient method for solving magnetic differential equations. The presence of islands and stochastic regions raises a host of new issues, which are the subject of continuing research.

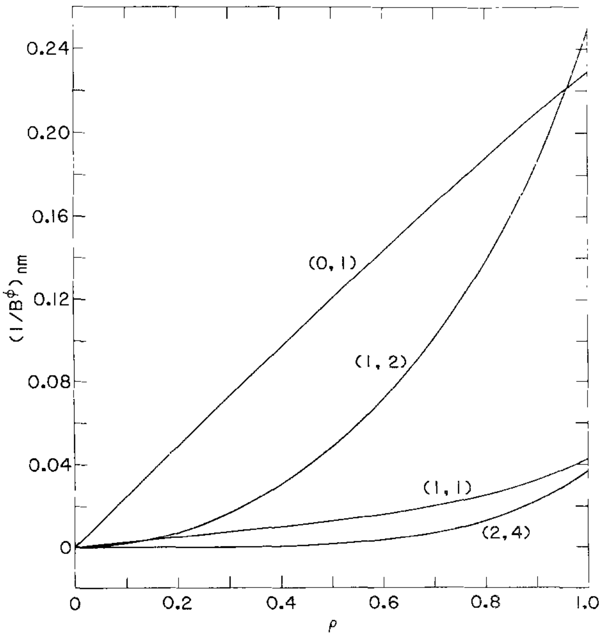

FIG. 2. The four largest Fourier modes in the transform of $1/B^{\phi}$, plotted as a function of ρ. The curves have been labeled with the corresponding mode numbers (n, m).

APPENDIX A: Estimates for Numerical Errors

In this appendix we outline our derivation of the estimates for numerical errors given in Section IV. We have functions of ϕ of the form of Eq. (17), and we wish to determine the coefficients of the Fourier modes. To handle the nonperiodicity we multiply by the window function $\exp[-(\phi/\phi_g)^2]$. To obtain the Fourier transform, we multiply by $\sin(k'\phi)$ (or $\cos(k'\phi)$), where $k'=\pi j/\phi_f$ for j an integer, and we integrate from $\phi=-\phi_f$ to $\phi=\phi_f$. Each Fourier mode of the function makes a contribution which is half the real part of

$$\int_{-\phi_f}^{\phi_f} \exp[-(\phi/\phi_f)^2+i(k-k')\,\phi]\,d\phi \mp \int_{-\phi_f}^{\phi_f} \exp[-(\phi/\phi_f)^2+i(k+k')\,\phi]\,d\phi, \quad \text{(A1)}$$

where k is the wave number of the mode, and where the $-$ and $+$ signs correspond to odd and even functions, respectively. Equation (A1) can be expressed in terms of error functions,

$$\sqrt{\pi}\phi_g \exp\{-[(k-k')\,\phi_g/2]^2\}\,\mathrm{erf}\left[\phi_f/\phi_g-\frac{i}{2}\,(k-k')\,\phi_g\right]$$
$$\mp\sqrt{\pi}\phi_g \exp\{-[(k+k')\,\phi_g/2]^2\}\,\mathrm{erf}\left[\phi_f/\phi_g-\frac{i}{2}\,(k+k')\,\phi_g\right]. \quad \text{(A2)}$$

To estimate the value of the integrals we use the asymptotic expansion of the error function, valid for ϕ/ϕ_g large,

$$\mathrm{erf}(z)\sim 1-\exp(-z^2)/\sqrt{(\pi)}\,z.$$

The dominant error comes from the $k-k'$ terms. For $k=k'$ we recover the normalization necessary to evaluate the Fourier coefficient, and we obtain the relative error due to finite integration length, Eq. (21). Dividing the value for $k\neq k'$ by that for $k=k'$, we determine the relative error due to spectral leakage, Eq. (22).

APPENDIX B: Table

The Most Important Numerical Parameters Discussed in This Paper, listed in the order in which they are defined in the paper

Parameter	Definition
$\mathcal{M}$	Maximum poloidal mode number (input)
$\mathcal{N}$	Maximum toroidal mode number per period (input)
L	$L+1$ is number of coordinate surfaces (input)
N	Number of periods (input)
maxexp	Maximum radial power pulled out of Fourier coefficient (input)
ϕ_f	Distance for which field lines followed (internal)
ϕ_g	Width of gaussian window used for Fourier transformation (internal)
ftprec	Accuracy of Fourier coeffs (input)

Note. After each we have specified whether it must be given to the code as input or is calculated internally from the input parameters.

Acknowledgments

We would like to thank A. Boozer for numerous helpful conversations. This work was supported by the U.S. Department of Energy under Contract DE-AC02-76-CHO-3073.

References

1. The phrase "magnetic differential equation" first appears in M. D. Kruskal and R. M. Kulsrud, *Phys. Fluids* **1**, 253 (1958). For a more general discussion of magnetic differential equations see Ref. [2].
2. L. S. Solov'ev and V. D. Shafranov, in *Reviews of Plasma Physics*, edited by M. A. Leontovich (Consultants Bureau, New York, 1970), Vol. V.
3. A. Reiman and H. Greenside, *Comput. Phys. Commun.* **43**, 157 (1986).
4. A. H. Boozer, *Phys. Fluids* **25**, 520 (1982).
5. G. Kuo-Petravic, A. H. Boozer, J. A. Rome, and R. H. Fowler, *J. Comput. Phys.* **51**, 261 (1983).
6. F. Bauer, O. Betancourt, and P. Garabedian, *Magnetohydrodynamic Equilibrium and Stability of Stellarators* (Springer-Verlag, New York, 1984).
7. S. P. Hirshman and D. K. Lee, *Comput. Phys. Commun.* **39**, 161 (1986).
8. U. Schwenn, *Comput. Phys. Commun.* **31**, 167 (1984).
9. S. Hamada, *Nucl. Fusion* **1**, 23 (1962).
10. J. M. Greene and J. L. Johnson, *Phys. Fluids* **5**, 510 (1962).
11. C. Gourdon, Rep. EUR-CEA-FC 413, 1966 (unpublished); C. Gourdon, G. Lemarie, F. Roche, and J. L. Soulé, Rep. EUR-CEA-FC 449, 1968 (unpublished).
12. *NAG Fortran Library Manual Mark 9* (Numerical Algorithms Group, Mayfield House, U.K., 1982).
13. C. deBoor, *A Practical Guide to Splines* (Springer-Verlag, New York, 1978).
14. O. Buneman, *SIAM J. Sci. Stat. Comput.* 7 (2), 624–638 (1986).
15. F. J. Harris, *Proc. IEEE* **66**, 51 (1978).
16. W. Gautschi, in *Handbook of Mathematical Functions*, edited by M. Abramowitz and I Stegun (National Bureau of Standards, Washington, DC, 1964), p. 295.
17. J. F. Lyon, B. A. Carreras, K. K. Chipley, M. J. Cole, J. H. Harris, T. C. Jernigan, R. L. Johnson, V. E. Lynch, B. E. Nelson, J. A. Rome, J. Sheffield, and P. B. Thompson, *Fusion Technol.* **10**, 179 (1986).
18. H. Greenside and A. Reiman, Sherwood Conference, San Diego, April, 1987 (unpublished).