JOURNAL OF COMPUTATIONAL PHYSICS 93, 411–425 (1991)

An Inverse Coordinate Multigrid Method for Free Boundary Magnetohydrostatics

P. S. CALLY

Department of Mathematics,
Monash University, Clayton, Victoria, Australia 3168

Received September 6, 1989; revised December 8, 1989

The equations of 2D magnetohydrostatic equilibrium with free (pressure) boundary conditions are expressed in inverse (i.e., flux) coordinates for both cartesian and axisymmetric geometries. The resulting quasi-linear elliptic system is solved using FAS full multigrid with line relaxation as the smoothing procedure. If field line connectivity is specified in the ignorable coordinate (i.e., field shear or twist is given), the system is governed by integro-differential equations, which are solved in the same way. Convergence rates are generally excellent, though an expanding fluxtube model, which provides a particularly difficult test, results in somewhat slower, though still acceptable, convergence. © 1991 Academic Press, Inc.

1. INTRODUCTION

Magnetic fields of varying scales, strengths, and geometries are ubiquitous in the solar atmosphere. They may emerge from the surface most notably in active regions (often associated with sunspots) or, on a smaller scale, from between supergranules. The structures they form in the atmosphere, e.g., loops, arcades, prominences, the network, canopies, coronal holes, etc., are often long lived (in relation to the Alfvén crossing time), and so static models may be appropriate. In some of these examples, in particular the network and canopies, the shape of the emerging magnetic structure is probably determined by a pressure balance across the boundary between the magnetic region and the surrounding (comparatively) field free plasma (Fiedler and Cally [1]). A free boundary problem results.

The Grad–Shafranov (G-S) equation, which governs two-dimensional magnetohydrostatic equilibrium, is a second-order nearly linear uniformly elliptic partial differential equation which may easily be solved using standard numerical techniques in the case of fixed boundary conditions. However, the free boundary problem is perhaps best formulated in terms of inverse or flux coordinates in which the magnetic field lines become coordinate lines. A second reason for such a transformation is that the energy equation, which one may wish to solve simultaneously with the G-S equation, reduces to ODEs along field lines when the radiation is optically thin (see [1]). (However, if radiative transfer effects are important, the

0021-9991/91 $3.00

field lines are no longer thermally isolated and the energy equation can not be simplified in this way.) Finally, if steady plasma flow is allowed, Alfvén's theorem guarantees that there can be no cross-field component of velocity, i.e., that the fluid flows along field lines. These three considerations argue strongly in favour of a method based on magnetic field coordinates.

In the literature, most magnetohydrostatic problems concern space filling field structures and therefore do not involve free boundaries. They are generally solved inside rectangular computational boxes, with the boundaries placed sufficiently far away from the region of interest that the necessarily rather artificial boundary conditions have little effect on the solution. Of the large number of studies of this type, we mention the magnetofrictional method of Yang *et al.* [2] and Klimchuk *et al.* [3], which is a pseudo-transient method based on Clebsch variables for the calculation of force-free 3D equilibria, and the more general technique of Zwingmann [4] which applies Newton iteration based on an energy principle.

Numerical solution of the free boundary problem without the use of flux coordinates has been carried out by Steiner *et al.* [5], who solve for emerging fluxtubes in the original cylindrical coordinates. However, this procedure is rather messy as it involves taking explicit account of the current sheet resulting from the discontinuity of the magnetic field on the boundary. A sort of "halfway house" is afforded by boundary fitted coordinate transformation techniques in which the boundaries are mapped to coordinate lines but the internal field lines (or stream lines etc.) are not (generally). These methods have been widely used in many contexts; see Thompson *et al.* [6] for a detailed review, and Hunt [7] and Pizzo [8] for recent examples which incorporate multigrid solution. Pizzo's discussion is of particular relevance here as it addresses the magnetostatic fluxtube problem. Though quite effective and comparatively easy to implement, this approach does not supply the further advantages mentioned above regarding the energy and flow equations.

Fiedler and Cally [1] implement what we shall choose to call a semi-inverse method in 2D, in which all field lines become coordinate lines, but where the second coordinate remains the usual cartesian variable z. Solution of the resulting partial differential equation is carried out using non-linear over-relaxation. The energy and flow equations are also incorporated.

In the formulation presented in this paper, the transformed G-S equation together with an accompanying arclength condition form a complicated quasilinear system beyond the capabilities of most of the standard fast elliptic solvers. One method which remains viable though is full approximation storage (FAS) multigrid (Brandt [9]). (A somewhat different inverse coordinate multigrid method for MHD was suggested by Braams [10], but not programmed.) We implement this in the full multigrid (FMG) mode for both axisymmetric and 2D cartesian models, and achieve generally satisfactory convergence rates.

There has also been some interest in calculating the structure of twisted flux tubes where the total amount of twist along field lines is specified. In 3D dynamical simulations (Steinolfson and Tajima [11]) this is treated straightforwardly, but for time independent 2D calculations it leads to a complicated non-linear integro-differential equation with an integro-differential boundary condition (if pressure is prescribed on the boundary). A somewhat different semi-inverse flux coordinate approach has been applied by Zweibel and Boozer [12], but solution was restricted to the small twist case. Using multigrid applied to our formulation this restriction is dropped and a representative large twist solution calculated.

2. Inverse Coordinate Formulation

In this paper, attention is focused on two-dimensional (2D) magnetohydrostatic equilibrium, both cartesian and axisymmetric, with the pressure boundary condition

$$p + p_{\mathrm{mag}} = p_{\mathrm{ext}}, \tag{1}$$

where p is the plasma pressure inside the field, $p_{\mathrm{mag}} = B^2/2\mu$ is the magnetic pressure expressed in terms of the field strength B, and p_{ext} is an imposed external pressure. The constant $\mu = 4\pi \times 10^{-7}$ H m^{-1} is the permeability of free space.

2.1. *Cartesian Formulation*

In the cartesian scenario, with x the ignorable coordinate, the magnetic field $\mathbf{B}$ can be expressed in terms of a potential A,

$$\mathbf{B}(y, z) = \left(b, \frac{\partial A}{\partial z}, -\frac{\partial A}{\partial y}\right). \tag{2}$$

Note that $\nabla \cdot \mathbf{B} = 0$ automatically (as required in the absence of magnetic monopoles) and $\mathbf{B} \cdot \nabla A = 0$, meaning that A is constant on field lines and so may be used to label them. The steady equation of momentum conservation balances pressure gradient, Lorentz, and gravitational forces,

$$\mathbf{0} = -\nabla p + \frac{1}{\mu}(\nabla \times \mathbf{B}) \times \mathbf{B} + \rho \mathbf{g}, \tag{3}$$

where ρ is the mass density and $\mathbf{g} = -g\hat{\mathbf{z}}$ is the gravitational acceleration. Substituting (2) into (3) and taking x, y, and $\hat{\mathbf{B}}$ components, it follows that

$$b = b(A), \tag{4a}$$

which states that b is constant on field lines,

$$\left(\frac{\partial p}{\partial z}\right)_A = -\rho g, \tag{4b}$$

from which the gravitational stratification of the internal plasma may be derived, and

$$\nabla_1^2 A = -\frac{d\lambda}{dA} - \mu\left(\frac{\partial p}{\partial A}\right)_z, \tag{4c}$$

where $\lambda = \frac{1}{2}b^2$ (see Fiedler and Cally [1]) and $\nabla_1^2 = \partial^2/\partial y^2 + \partial^2/\partial z^2$ is the two-dimensional Laplacian. The subscripts A and z on the partial derivatives indicate that these variables, respectively, are held constant. The elliptic equation (4c) is called the Grad–Shafranov equation, and is linear except for the terms on the right-hand side. For the moment, regard $\lambda(A)$ and $p(A, z)$ as known functions.

In order to treat the free boundary, it is found convenient to introduce coordinates based on the field lines. If $A = 0$ and $A = A_{\max}$ represent the two bounding field lines, the coordinate $u = A/A_{\max}$ is used. The second coordinate v, defining position along each field line, need not be specified yet, but in general is given by some differential relation

$$F(u_y, u_z; v_y, v_z; u, v) = 0. \tag{5}$$

For example, orthogonal coordinates are generated by $u_y v_y + u_z v_z = 0$. In inverse coordinates, instead of solving for $u(y, z)$ and $v(y, z)$, the roles of dependent and independent coordinate are reversed and we seek $y(u, v)$ and $z(u, v)$. With $J = y_u z_v - y_v z_u$ being the Jacobian, it is easily shown that

$$u_y = \frac{z_v}{J}, \qquad u_z = -\frac{y_v}{J}, \qquad v_y = -\frac{z_u}{J}, \qquad v_z = \frac{y_u}{J}. \tag{6}$$

Hence the Grad–Shafranov equation takes the form

$$\begin{aligned} J^{-3}\{ & y_v[(y_v^2 + z_v^2)\, z_{uu} - 2(y_u y_v + z_u z_v)\, z_{uv} + (y_u^2 + z_u^2)\, z_{vv}] \\ & - z_v[(y_v^2 + z_v^2)\, y_{uu} - 2(y_u y_v + z_u z_v)\, y_{uv} + (y_u^2 + z_u^2)\, y_{vv}]\} \\ = {} & -(\mu(p_u)_z + \lambda_u)/A_{\max}^2, \end{aligned} \tag{7}$$

where $(p_u)_z$ represents $\partial p/\partial u$ at constant z. Together with the inverse form of (5), this comprises a non-linear system of differential equations for y and z.

Orthogonal coordinates are not ideal for our purpose. In inverse coordinates their defining equations becomes $y_u y_v + z_u z_v = 0$, which is strongly non-linear and numerically troublesome (although it does dispense with the cross derivatives in (7)). Furthermore, since field lines do not generally emerge from or re-enter the solar surface vertically, these boundaries will not be coordinate lines. One simple possibility is to set $v = z$ which recovers the semi-inverse method of Fiedler and Cally [1]; however, this is inappropriate if the field turns over or becomes nearly horizontal, and we shall not consider it further here. Instead, it is generally convenient to use normalized projected arc length s defined by $ds = (y_v^2 + z_v^2)^{1/2}\, dv$; i.e., set $v=0$ and $v=1$ at the ends of each field line (e.g., where it leaves and re-enters the surface for a returning flux model) and require $d^2s/dv^2=0$. Thus

$$y_v y_{vv} + z_v z_{vv} = 0. \tag{8}$$

Although this generally produces good results, it is obvious that for some magnetic configurations it will lead to the u and v coordinate lines crossing at a fine angle, which both reduces resolution and interferes with convergence. In some cases this can be avoided by skewing the end boundaries, but if that is not feasible we must be prepared to either accept the slower convergence, or adopt a different prescription for the v coordinate (see Thompson *et al.* [6] for a discussion of the possibilities).

A simple means of coordinate stretching in the v direction is obtained by replacing (8) with

$$y_v y_{vv} + z_v z_{vv} = D(v)(y_v^2 + z_v^2),$$

where D is chosen to provide enhanced resolution in part of the domain. In fact, we may even set $D=D(u,v)$, thus achieving substantial control over the coordinate system used. However, convergence is adversely affected if the stretching is too severe, in which case local grid refinement should be used instead (Brandt [13]).

The side boundary ($u=0$ and $u=1$) pressure condition in inverse coordinates is

$$\frac{y_v^2 + z_v^2}{J^2} = \frac{2\mu(p_{\text{ext}} - p) - 2\lambda}{A_{\max}^2}, \tag{9}$$

which is also non-linear. On the ends, $v=0$ and $v=1$, various possibilities present themselves, but the most obvious and easiest to implement is the Dirichlet condition where (y, z) is specified. We shall adopt this throughout, but field angle or constant total pressure conditions are also possible.

2.2. *Axisymmetric Formulation*

Assuming that the angle θ in cylindrical coordinates (r, θ, z) is ignorable, the magnetic field may be expressed in terms of a scalar potential α:

$$\mathbf{B} = \frac{1}{r}\left(-\frac{\partial\alpha}{\partial z}, \beta, \frac{\partial\alpha}{\partial r}\right). \tag{10}$$

It turns out that both α and β are constant on field lines. With $\lambda = \frac{1}{2}\beta^2$ the Grad–Shafranov equation becomes

$$r\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial\alpha}{\partial r}\right) + \frac{\partial^2\alpha}{\partial z^2} = -\frac{d\lambda}{d\alpha} - \mu r^2 \left(\frac{\partial p}{\partial\alpha}\right)_z. \tag{11}$$

The hydrostatic equation (4b) is unchanged.

We now convert to inverse coordinates, with $u=\alpha/\alpha_{\max}$ and the projected normalized arclength v being the independent variables. As before the height z will be used as a dependent variable. The radial distance r is not a suitable choice as the second dependent variable because $r \sim u^{1/2}$ in the neighbourhood of the axis, making the standard finite difference representation of r_u very inaccurate for the first few grid points. Instead, we shall use $\varpi = r^2$, which avoids this difficulty. The equations are then

$$\begin{aligned} J^{-3}\{&\varpi_v[(\varpi_v^2+4\varpi z_v^2)\, z_{uu} - 2(\varpi_u\varpi_v + 4\varpi z_u z_v)\, z_{uv} + (\varpi_u^2 + 4\varpi z_u^2)\, z_{vv}] \\ &- z_v[(\varpi_v^2+4\varpi z_v^2)\, \varpi_{uu} - 2(\varpi_u\varpi_v + 4\varpi z_u z_v)\, \varpi_{uv} + (\varpi_u^2+4\varpi z_u^2)\, \varpi_{vv}]\} \\ &= -(\mu\varpi(p_u)_z + \lambda_u)/\alpha_{\max}^2, \end{aligned} \tag{12}$$

$$\frac{\varpi_v\varpi_{vv}}{4\varpi} - \frac{\varpi_v^3}{8\varpi^2} + z_v z_{vv} = 0, \tag{13}$$

and

$$\frac{\varpi_v^2 + 4\varpi z_v^2}{J^2} = \frac{2\mu\varpi(p_{\text{ext}} - p) - 2\lambda}{\alpha_{\max}^2} \quad \text{on} \quad u = 1. \tag{14}$$

The Jacobian is now $J = \varpi_u z_v - \varpi_v z_u$.

2.3. *Twist and Shear Constraints: Integro-Differential Equations*

Till now, λ has been assumed to be given. However, physically, it is often more realistic to specify the total twist (cylindrical geometry) or shear (cartesian), or in other words, to completely specify the field line connectivity from $v=0$ to $v=1$. For example, if we twist one end of a previously untwisted cylinder through an angle Φ, we know which points connect to which on the two ends, but not the value of $\beta = rB_\theta$ on the field lines.

Let $L(u)$ be the total x-displacement along the field line u in the cartesian model. It is a simple matter to show that

$$\lambda = \frac{L^2 A_{\max}^2}{2[\int_0^1 J\, dv]^2}. \tag{15}$$

Similarly, in cylindrical geometry, let $T(u) = \Phi/2\pi$ be the total number of twists along u. Then,

$$\lambda = \frac{8\pi^2 T^2 \alpha_{\max}^2}{[\int_0^1 J/\varpi\, dv]^2}. \tag{16}$$

On substituting these into the appropriate Grad–Shafranov equation ((7) or (12)) and the pressure boundary condition ((9) or (14)), a complicated non-linear integro-differential problem results.

3. Numerical Procedure

For the reader expert in the standard multigrid concepts and terminologies (Brandt [9, 13]; Stüben and Trottenberg [14]), we shall proceed immediately to discuss only the details of the present implementation. However, a brief review is provided in the Appendix for those less familiar with the method.

Equations (7) and (8) (or (12) and (13)) are discretized using standard second order centered differences, whilst the u derivatives in the boundary condition (9) (or (14)) are expressed using the second-order three point one sided difference. Coarsening is in both the u and v directions. The grids used are labelled $K=0, 1, ..., M$, progressing from coarsest to finest. With N_u^K and N_v^K denoting the number of intervals on the Kth grid in the u and v directions, respectively, a representative mesh point (u_i^K, v_j^K) on that grid is given by $(i/N_u^K, j/N_v^K)$, $i=0, 1, ..., N_u^K$, $j=0, 1, ..., N_v^K$.

Bilinear interpolation is employed within cycles, but bicubic interpolation was found necessary between cycles in the FMG algorithm. Restriction uses full weighting. Both V and W cycles (with a limited adaptive facility) have been tried; the V cycle with $\nu_1=\nu_2=2$ (these are the number of *pre* and *post* smoothing sweeps carried out before descending to a coarser level and after ascending to a finer one, respectively) usually proves to be the most efficient, though this depends upon the particular case being run; sometimes the $\nu_1=\nu_2=1$ W cycle is marginally quicker. In any case, convergence in under 10 WU is often achieved no matter which (reasonable) combination of multigrid parameters is adopted (1 WU, or work unit, is the amount of work required for one sweep on the finest grid). Convergence of difficult models can be slower though (see Sections 4.1 and 4.2).

Local mode analysis (Brandt [9]) of Eqs. (7) and (8) suggests that point Gauss–Seidel is a poor error smoother if $\mathscr{A}=(y_u^2+z_u^2)\,h_u^2$ and $\mathscr{B}=(y_v^2+z_v^2)\,h_v^2$ differ substantially in magnitude, where h_u and h_v are the grid spacings in the u and v directions, respectively. The geometric interpretation of these two terms is clear; they are respectively the squares of the lengths, in physical space, of the $v=\text{constant}$ and $u=\text{constant}$ sides of a grid element. In other words, long thin grid elements (even in the ratio 3 : 1, say) lead to poor convergence rates. On the other hand, a similar analysis of lexographic Gauss–Seidel horizontal line relaxation (i.e., relaxing a whole line of constant v simultaneously) shows it to be a good smoother even if $\mathscr{A} \ll \mathscr{B}$ (long or closely packed field lines), whilst vertical line relaxation performs well for $\mathscr{B} \ll \mathscr{A}$ (short or widely spaced field lines). Both are satisfactory when $\mathscr{A} \sim \mathscr{B}$. Zebra ordering is found to be inferior for this problem. For models in which both regimes exist, alternating direction line relaxation is performed (i.e., a horizontal sweep followed by a vertical sweep; greater efficiency could be obtained by only performing the required sweep locally). The pressure boundary condition is both appended to the ends of the horizontal sweep and applied as a separate vertical sweep, as required. Of course, these sweeps are performed on the linearized version of the equations, and involve the solution of hex- or septdiagonal systems for horizontal and vertical line relaxation, respectively. In all cases, it is found best to apply only one Newton iteration per sweep. The solution on the coarsest grid $K=0$ is obtained by simple repeated application of the relaxation procedure till convergence is obtained; although this may require many sweeps, it is cheap due to the small number of points on that mesh.

The integro-differential systems for fields with imposed shear or twist discussed in Section 2.3 are solved using line relaxation along field lines, where the integrals $\int_0^1 \cdots dv$ are discretized using the trapezoidal rule. The resulting matrix equations may still be solved directly in an efficient manner.

Errors are monitored by calculating the RMS change in position of the internal points after each relaxation sweep (boundary points are monitored separately). Define the "error" after a sweep on the Kth grid in the cartesian geometry (and similarly for the axisymmetric case) by

$$E=\left\{\frac{1}{(N_u^K-1)(N_v^K-1)}\sum_{i=1}^{N_u^K-1}\sum_{j=1}^{N_v^K-1}[(\delta y_{ij})^2+(\delta z_{ij})^2]\right\}^{1/2}. \tag{17}$$

Within the FMG algorithm, convergence is deemed to have been reached at a particular level K when the last sweep in a (V or W) cycle yields $E<\varepsilon$, and similarly for the boundary error, where ε is some preset tolerance (10^{-5} in the cartesian test problems, 10^{-4} for the cylindrical models). Overall convergence occurs when the finest level $K=M$ converges. The limited adaptive facility mentioned above involves: (i) performing less than ν_1 or ν_2 sweeps on coarse levels within a cycle when the error on that level has already been reduced to less than ε, though the full number of sweeps is always applied on that cycle's finest mesh; and (ii) truncating a cycle when the first ν_1 sweeps on its finest level have already reduced the error to below ε, provided that at least one cycle at that level has already been completed.

Because of the frequent jumping around between mesh levels in multigrid, it is not easy to define an unambiguously useful measure of convergence rate. One that is sometimes applied compares the error from the last sweep of the last cycle with that from the last sweep of the previous cycle (e.g., Hunt [7]). However, this is unsatisfactory when (as is usually the case) there is only one cycle on the finest level, since then we are comparing an error on $K=M$ with one on $K=M-1$. Properties of the interpolation method and differing relaxation rates on the two grids then intrude. Instead, we shall define a convergence factor on level K, $\varrho(K)$, to be the ratio of the error from the last sweep on the last cycle of that level to that from the first sweep of that same cycle. Although this slightly underestimates the improvement obtained in a cycle, because it does not include the first sweep, it is a more consistent measure than that mentioned above. However, the most meaningful measures of relative speed between different methods are the total number of work units (WU) and the cpu time (CPU) expended in the solution, though even these can be misleading when drawn from particular cases.

4. Test Problems and Results

Three test problems are considered, one in cartesian geometry and the other two cylindrical.

4.1. *Arcade*

A well-known (Priest [15]) exact 2D model for an evacuated arcade surrounded by an isothermal atmosphere with base pressure and scale height scaled to unity, $p_{\text{ext}} = e^{-z}$, is

$$\begin{aligned} b = B_x &= B_0\left(1 - \frac{1}{4k^2}\right)^{1/2} \cos ky\; e^{-z/2}, \\ B_y &= \frac{B_0}{2k} \cos ky\; e^{-z/2}, \\ B_z &= -B_0 \sin ky\; e^{-z/2}. \end{aligned} \tag{18}$$

where $B_0 = \sqrt{2\mu}$. The field lines are uniformly sheared in the x direction by an angle γ, $(\tan\gamma = B_x/B_y = \sqrt{4k^2 - 1})$. In general, this is a so-called constant-α field, becoming potential when the shear, γ, vanishes.

The test model, depicted in Fig. 1, has its inner field lines $(u = 0)$ anchored to the surface $z = 0$ at $\pm y_{\min} = \pm 1$ and its outer field lines $(u = 1)$ at $\pm y_{\max} = \pm 2$. The imposed shear is 30° $(k = 1/\sqrt{3})$. The end Dirichlet boundary conditions are

$$y = \frac{1}{k}\cos^{-1}[\cos ky_{\min} - (\cos ky_{\min} - \cos ky_{\max})\, u] \tag{19a}$$

and

$$z = 0 \tag{19b}$$

on $v = 0$ and $v = 1$.

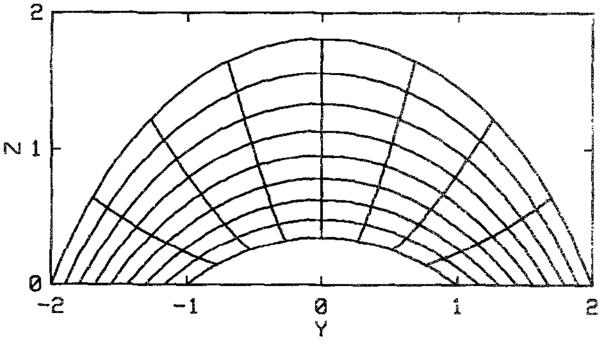

FIG. 1. Converged arcade model, showing field lines and lines of constant v. Only every 16th line used in the calculation is shown.

TABLE I

Arcade Model Accuracies and Convergence Rates

Mesh size	$z(1, 0.5)$	$z(0, 0.5)$	ϱ
4×4	1.73189	0.30853	
8×8	1.79552	0.34339	0.209
16×16	1.80805	0.35113	0.254
32×32	1.81029	0.35281	0.064
64×64	1.81124	0.35341	0.099
128×128	1.81156	0.35359	0.187
Exact	1.81173	0.35368	

Note. ϱ is the convergence factor (see text).

The finest grid is 128×128, with no allowance being made for symmetry (the program is written to solve for asymmetric arcades), thus roughly doubling the amount of work required in this test compared to what is strictly necessary. The figure shows every sixteenth field line and line of equal v, indicating that the elemental grid boxes are significantly longer along field lines than across them, especially towards the outside of the arcade. For this reason horizontal (i.e., cross-field) line relaxation is used. If, on the other hand, $y_{\min}$ is zero or close to it (filled arcade), the inner grid boxes become vanishingly short along field lines, necessitating the use of vertical (i.e., along field) relaxation.

As a measure of accuracy, we compare the arcade height $z(1, 0.5)$ and underside height $z(0, 0.5)$ calculated on various grids with their exact values (Table I). The $\nu_1=\nu_2=2$ V cycle with only horizontal line relaxation is used. The convergence factors, ϱ, as described above, refer to the last cycle on each level.

Four different methods are compared in Table II. The initial guess was the same in each case (the field lines being a sequence of elliptical arcs). The cpu times shown (CPU) were obtained on a serial machine, a DEC8700. Our calculation of WU takes into account only the relaxation sweeps and the restriction step, and so is an

TABLE II

Arcade Model Convergence for Various Multigrid Parameters

ν_1	ν_2	Vert. sweep	Cycle	WU	CPU	ϱ
2	2	No	V	7.06	1:04	0.187
1	1	Yes	V	8.10	1:18	0.064
2	2	No	W	8.80	1:21	0.119
1	1	No	W	6.09	1:00	0.264

Note. CPU is given in minutes and seconds. Horizontal sweeps are performed in all cases, but vertical sweeps only where indicated.

imperfect measure of work done; nevertheless it is probably accurate to within about 30%. The best performances in this test were produced by the $\nu_1 = \nu_2 = 2$ V cycle and the $\nu_1 = \nu_2 = 1$ W cycle, without vertical sweeps, but all four methods performed creditably.

4.2. *Expanding Fluxtube*

Determining the shape of variable cross-section pressure bounded flux tubes is a notoriously difficult problem (Browning and Priest [16, 17]). Indeed, to the best of the author's knowledge, there are no known exact solutions outside the rather artificial class in which a field is prescribed and the required bounding pressure calculated a posteriori. Numerical solution may be carried out in $r-z$ space (Steiner *et al.* [5]), or using the semi-inverse method (Fiedler [18]). The former has the disadvantage of requiring the consideration of current sheets, whilst the latter can lead to problems where the field lines becomes nearly horizontal. We shall apply the fully inverse method.

The example presented here concerns an untwisted tube expanding in response to a decreasing external pressure. We again set $p_{\text{ext}} = e^{-z}$. The specific problem is not particularly relevant physically, but is just chosen to illustrate the method. Consider two uniform flux tubes, the first of radius $R_L = 4$ and lying along the z-axis in $z < 0$, and the second of radius $R_U = 4e^{5/4} = 13.96$ in $z > 5$. Our task is to find the expanding tube which connects them. The magnitude of the expansion here is quite severe, making it a tough test. Using a 32×64 fine grid, $M = 3$, and a required accuracy of 10^{-4}, the solution depicted in Fig. 2 is obtained in 22.7 WU ($\nu_1 = \nu_2 = 1$ W cycle with vertical line relaxation only). Although slow for multigrid, this would be considered excellent by most standards. The relatively poor convergence is due to quite severe grid deformation in this model; in the less testing model $R_L = 1$, $R_U = e^{5/4} = 3.49$, convergence takes 13 WU.

This example is closest to the expanding flux tube models of Steiner *et al.* [5], though there are differences of detail, and so it provides one of the few checks of numerical efficiency available to us. A truly fair comparison could only be made by solving for the same model on the same machine, but a few general points can be made. The free boundary flux tube models of Steiner *et al.* converge to an accuracy

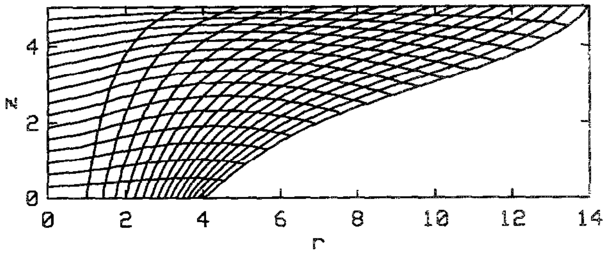

FIG. 2. Converged expanding fluxtube model (axially symmetric) calculated on a 32×64 grid. Note the large grid distortion near the top, which results in impaired convergence.

of 10^{-5} in around 75 of their iterations. Each such iteration consists of a number of quite complex steps involving intermediate evaluation of the current density, both internally and on the current sheet, and subsequent calculation of the field using Ampère's law. It would appear that each iteration must involve more work, perhaps considerably more work, than one of our relaxation sweeps, especially if the semi-inverse method is used. The tubes they consider expand relatively slowly, much more slowly than the one shown in Fig. 2. Such models are handled particularly easily by our method and would typically converge to the above accuracy in under 15 WU, i.e., under 15 equivalent relaxation sweeps on the fine grid. We might therefore expect the present method to have an efficiency advantage of at least a factor of 5, and probably much more.

4.3. *Twisted Fluxtube*

We consider an initially uniform fluxtube of length $L = 4$ and radius $R = 1$ contained by an external pressure $p_{\text{ext}} = 1$. A differential twist $T(u) = T_0 e^{-3u}$ with $T_0 = 2$ is applied at one end (corresponding to a Gaussian twist profile in r at the ends, since $u = r^2$ at $z = 0$ and L). The width of the tube then becomes non-uniform in order to maintain equilibrium. As explained in Section 2.3, the structure of the twisted tube may be found by solving a non-linear integro-differential equation. Our program is very inefficient for this model for two reasons. First, it does not take account of the symmetry about $z = L/2$. Second, the arclength variable v is unnecessary; the semi-inverse method which solves for $\varpi(u, z)$ would be quite adequate in this case. Nevertheless, the model serves to display the capabilities of the general method.

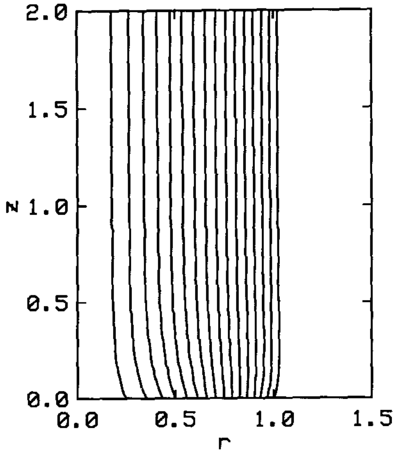

FIG. 3. Twisted fluxtube model (axially symmetric), showing only the lower half of the tube (there is symmetry about $z = 2$). The curves delineate flux surfaces rather than field lines. The twist along field line u, where $u = r^2$ on the base, is $T(u) = 2e^{-3u}$. Lines of equal v have been suppressed.

Figure 3 results from a fine grid of 32×128 with $M=3$, and $\nu_1 = \nu_2 = 1$ W cycles. Convergence to an accuracy of better than 10^{-4} (actually 1.1×10^{-5} internally and 4×10^{-5} on the boundary) is obtained in 8.5 WU. Models where λ is prescribed rather than T generally converge even more quickly. Decreasing T_0 also results in faster convergence; for example, $T_0 = 1$ in this model takes 6 WU.

5. Conclusions

Although expressing the equations of magnetohydrostatics in inverse coordinates makes them appear more complicated, it simplifies the free boundary and applied twist or shear problems considerably. Solution of the resulting non-linear partial differential or integro-differential equations is practical using FAS multigrid, though the complexity of the equations results in a high operation count per sweep. Improvements to the methods used in this paper may result from (a) better smoothing procedures, (b) more sophisticated treatment of the twist or shear integrals, and (c) modifications to the definition of the variable v in some geometries. With regard to the last of these, in many circumstances the choice $v = z$ (the semi-inverse method) will be quite adequate, and will certainly be easier to program and faster to run.

APPENDIX: Multigrid Review

In this appendix we give a brief and necessarily incomplete account of multigrid methods. References to full expositions may be found in the text, and the interested reader is urged to consult them. However, for those with only a passing interest in the details of the numerical procedures used in this paper, the fundamental concepts necessary to understand Section 3 are summarized here.

The basic idea behind multigrid hinges on the concept of error smoothing. Let h be the grid spacing employed in finite differencing a 2D elliptic partial differential equation (for simplicity assume this to be the same in both directions). The convergence rate R of iterative methods depends upon h; for example, Gauss–Seidel iteration gives $R \sim h^2$ (Ames [19]). This can become prohibitively slow if h is small. Now, as shown by Brandt [9], the various Fourier modes of the error are reduced at different rates, with the long wavelength errors being responsible for the slow convergence. On the other hand, short wavelength errors (between $2h$ and $4h$ in size) are damped very rapidly, typically by an order of magnitude in around three sweeps of the iterative procedure. For this reason, we refer to Gauss–Seidel or other iteration procedures as *smoothers*.

Basic multigrid starts with an approximation on a fine grid, performs ν_1 smoothing sweeps, then *restricts* the solution to the coarser grid $2h$ using either straight injection or some weighting algorithm (actually, the problems solved on the various grids are slightly different, to take account of the discretization errors, but this does not affect the basic concepts). Then ν_1 further sweeps are made which damp errors between $4h$ and $8h$. There are only one quarter as many grid points on this level, so the sweeps are quite cheap. We may then, of course, proceed to still coarser and cheaper grids if we wish; generally it is efficient to go right down to, say, a 3×3 or 4×4 level where the problem is solved exactly or to some stringent accuracy. In this way the troublesome long wavelength errors on the fine grid are greatly reduced.

We must next return to the finest grid. This is done by stages, according to one of a number of possible strategies. For example, in the *V cycle* algorithm, we interpolate up to the second coarsest grid, perform ν_2 smoothing sweeps to damp errors introduced by the interpolation, then move up to the next level, smooth again, and so on. Thus, if we are using say three levels in all, labelled 2 (finest), 1, and 0, one V cycle consists of the following sequence: 2–1–0–1–2. An alternative procedure is the *W cycle*, which, for the above example, consists of 2–1–0–1–0–1–2. A number of such cycles may be necessary to achieve convergence.

Full multigrid (FMG) involves starting with an initial guess on the coarsest rather than finest level and iterating to convergence (or solving directly). This is an efficient way to quickly improve a poor first approximation. We then interpolate up to Level 1 and perform as many multigrid cycles as are required for convergence on that grid. Interpolation to Level 2 follows, where we again cycle to convergence, and so on till the finest level is reached and ultimate convergence has been achieved.

Acknowledgment

Much of the work for this paper was carried out while the author was a guest of the Mathematical Sciences Department of the University of St. Andrews. He would particularly like to thank Bernie Roberts, Eric Priest, and Alan Hood for their hospitality and assistance.

References

1. R. A. S. Fiedler and P. S. Cally, *Solar Phys.* **126**, 69 (1990).
2. W. H. Yang, P. A. Sturrock, and S. K. Antiochos, *Astrophys. J.* **309**, 383 (1986).
3. J. A. Klimchuk, P. A. Sturrock, and W. H. Yang, *Astrophys. J.* **335**. 456 (1988).
4. W. Zwingmann, *Solar Phys.* **111**, 309 (1987).
5. O. Steiner, G. W. Pneuman, and J. O. Stenflo, *Astronom. Astrophys.* **170**, 126 (1986).
6. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *J. Comput. Phys.* **47**, 1 (1982).
7. R. Hunt, *J. Comput. Phys.* **65**, 448 (1986).
8. V. J. Pizzo, in *Proceedings of the Chapman Conference on The Physics of Magnetic Flux Ropes, Bermuda, 1989.*
9. A. Brandt, *Math. Comput.* **31**, 333 (1977).
10. B. J. Braams, in *Multigrid Methods II, Proceedings, Cologne 1985*. Lecture Notes in Mathematics, Vol. 1228, edited by W. Hackbusch and U. Trottenberg (Springer-Verlag, Berlin, 1986), p. 38.
11. R. S. Steinolfson and T. Tajima, *Astrophys. J.* **322**, 503 (1987).
12. E. G. Zweibel and A. H. Boozer, *Astrophys. J.* **295**, 642 (1985).
13. A. Brandt, in *Multigrid Methods, Proceedings, Köln–Porz. 1981*, Lecture Notes in Mathematics, Vol. 960, edited by W. Hackbusch and U. Trottenberg (Springer-Verlag, Berlin, 1982), p. 220.

14. K. STÜBEN AND U. TROTTENBERG, in *Multigrid Methods, Proceedings, Köln-Porz, 1981*, Lecture Notes in Mathematics, Vol. 960, edited by W. Hackbusch and U Trottenberg (Springer-Verlag, Berlin, 1982), p. 1.

15. E. R. PRIEST, *Solar Magnetohydrodynamics* (Reidel, Dordrecht, 1982), p. 138.

16. P. K. BROWNING AND E. R. PRIEST, *Geophys. Astrophys. Fluid Dyn.* **21**, 237 (1982).

17. P. K. BROWNING AND E. R. PRIEST, *Astrophys. J.* **266**, 848 (1983).

18. R. A. S. FIEDLER, Ph.D. thesis, Monash University, 1989

19. W. F. AMES, *Numerical Methods for Partial Differential Equations* (Academic Press, New York, 1977), p. 117.