CLM - R 166

CLM - R 166

UKAEA

Report

THE COMPUTATION OF MULTIPOLE MHD EQUILIBRIA IN AXISYMMETRIC AND STRAIGHT GEOMETRY

DO NOT
REMOVE!

C Ll THOMAS

CULHAM LABORATORY
Abingdon Oxfordshire

1979

C/18 Available from H. M. Stationery Office £1.50

CLM-R 166

THE COMPUTATION OF MULTIPOLE MHD EQUILIBRIA IN AXISYMMETRIC AND STRAIGHT GEOMETRY

by

C.Ll. Thomas

Culham Laboratory, Abingdon, Oxon, OX14 3DB, UK
(Euratom/UKAEA Fusion Association)

Abstract

The details of the numerical methods used in codes for computing MHD equilibria in discrete conductor configurations are described with both code users and code writers in mind. Results produced by the codes have been successfully verified against analytic results and independent computations. The axisymmetric code has proved to be a valuable diagnostic aid for the TOSCA experiment. The user images of the codes are described in the appendices.

January 1979

SBN 85311 078 6

1. INTRODUCTION

In [1], Thomas and Haas described a number of axisymmetric MHD equilibrium computations in which either the plasma boundary shape was a specified flux surface or the plasma was surrounded by a vacuum region which was contained within a conducting shell of specified shape. The latter computations were relevant to experiments in which the equilibrium was provided by a conducting shell. Most present and proposed tokamak experiments have no copper shell and the vertical field for equilibrium is provided by currents in discrete conductors. An iron core may also be present but its vertical field can always be simulated numerically by an appropriate set of conductors. The absence of the shell allows windings to be used to shape the plasma cross section and also allows divertors to be operated. There is no natural computational boundary to the problem and hence the method of solution becomes rather more complex than in our earlier calculations.

In this report, we describe 2 codes for computing multipole MHD equilibria; one code treats the axisymmetric problem and the other treats the straight problem. These codes have been under continuous development for several years and during this time, a number of publications [2,3,4,5,6] on this subject have appeared. Where suitable, we have incorporated any new ideas from these publications into our codes. The numerical methods and facilities of the codes are described in Section 2. In Section 3, we verify the results of the codes with analytical work of Strauss [7], Papaloizou et al [8] and Mukhovatov and Shafranov [9]. The use of the codes as a diagnostic aid for the TOSCA [10,11] experiment is referred to in Section 4. Finally, in the appendices we give the user images of the codes.

2. NUMERICAL METHODS FOR COMPUTING MULTIPOLE EQUILIBRIA

Numerical methods for computing axisymmetric MHD equilibria with scalar pressure have received considerable attention recently. The first published computations were by Callen and Dory [12]. They considered a plasma which completely filled a conducting shell and which had a current density which was nonlinear in the poloidal flux function ψ. The finite difference equations were solved by successive overrelaxation (SOR). A configuration in which the plasma was confined away from a conducting wall by a limiter was studied by Chu et al.[4]. They used Picard's method and solved the linear system at each step by SOR. At each Picard step, the plasma boundary was defined as the last closed surface which touched the limiter. The current density contained a scale factor which was adjusted at each step so that the total plasma current was some specified value. The first application of constraints to the calculation of axisymmetric MHD equilibria was in fact made by Feneberg and Lackner [6]. They considered multipole

equilibria by using Picard's method with a numerical evaluation of the analytic solution of the equations at each step. In their paper, they pointed out that whenever a trivial solution was admissible, Picard's method without constraints would always converge to this trivial solution. It was also stated that their semi-analytic method was inapplicable to a straight system. This is because their grid system extended to infinity and whereas in an axisymmetric system the stream function $\psi(= RA_\phi)$ tends to zero at infinity, in a straight system $\psi(= A_z)$ tends to infinity. Here R is the distance from the axis of symmetry and A_ϕ is the azimuthal component of the vector potential in a toroidal system and A_z is the longitudinal component of the vector potential in a straight system.

It should be pointed out that the multipole equilibrium problem always admits a trivial solution. The typical formulation is

$$L\psi = G(\psi) \quad \text{in the plasma, where} \quad \psi < \psi_p$$
$$L\psi = 0 \quad \text{in the vacuum, where} \quad \psi > \psi_p$$

with some appropriate boundary conditions. Here L is an operator determined by the geometry, $G(\psi)$ is determined by the plasma currents and ψ_p is the flux at the plasma boundary. A mathematically acceptable solution is given by the solution of $L\psi = 0$. Since $\psi > \psi_p$, the solution has no knowledge of the first equation and hence there is no plasma. This is the trivial solution.

Our approach to computing multipole MHD equilibria with axisymmetry is not dissimilar from the approaches of Chance et al [13], Lackner [5], and Cennacchi et al. [2]. However, for theoretical purposes we have also provided a code which computes multipole equilibria in straight geometry in an analogous fashion.

The user is required to specify certain data. Typically these will be details of the external conductors, the position of the grid, the functional form of the plasma current density, the total plasma current, the position of the limiter and parameters related to the iterative method of solution.

We now consider each stage of the solution procedure in detail.

2.1 Specification of the Initial Data

A rectangular grid is placed with its boundary between the conductors and the expected position of the plasma boundary. Typically the position of the vacuum vessel will be a guide to the position of the grid, which need not be square.

The appropriate external filamentary coils are set up. In the axisymmetric case, the coils must be axisymmetric and in the straight case the coils also must be straight. In other words the coil currents must be in the direction of

the main plasma current. Filamentary coils are standard in these two codes so finite size conductors must be represented by a number of filaments. In both cases, the stream function due to a single filament is given by an analytic formula which is evaluated by standard library subroutines. In the axisymmetric case the stream function at the point (r,z) due to a current I flowing in a loop radius r_o and a distance z_o from the mid plane is

$$\psi = rA_\phi = \frac{\mu_o I}{\pi}\left(\frac{r_o r}{k^2}\right)^{\frac{1}{2}}\left[\left(1 - \frac{k^2}{2}\right)K(k) - E(k)\right]$$

where

$$k^2 = \frac{4r_o r}{(r_o + r)^2 + (z - z_o)^2}$$

and $\mu_o = 4\pi\ 10^{-7}$ henrys per metre. In the straight case, the stream function at the point (x,y) due to a current I flowing in a wire infinite in the z direction, passing through (x_o,y_o) is

$$\psi = A_z = - \frac{\mu_o I}{2\pi}\ \ell n\,[(x - x_o)^2 + (y - y_o)^2]^{\frac{1}{2}}$$

Some initial guess for the solution is made. The fine scale details of the initial guess do not appear to be important but a certain amount of care is necessary. Constraints will be discussed in Section 2.2 but the initial guess has to be considered along with these constraints. A tokamak contains a vacuum vessel and a limiter. The limiter defines the plasma boundary (in the absence of a separatrix) - it is the outermost surface which touches the limiter and should lie within the vacuum vessel. Hence the surfaces of the initial guess should be such that the initial plasma boundary lies within the vacuum vessel, with at least one point on the boundary defined by the limiter.

2.2 Application of Constraints

The application of constraints is an essential part of plasma-vacuum calculations. The constrained variables usually have precise physical interpretations. The main constraint is the geometrical constraint, where the plasma boundary is forced to touch some prescribed contour (limiter). No multipole tokamak equilibrium calculation has ever been performed without this constraint. Papaloizou et al [8] give a strong indication for the necessity to apply such a constraint. They show that a straight uniform current density plasma in an external quadrupole field has a certain height to width ratio, which is determined by the exterior field and the value of

the current density. However, in the absence of a geometrical constraint, the actual size of the plasma is arbitrary. An alternative viewpoint is that in general we have no information about the value of the flux at the plasma boundary. Application of the geometrical constraint removes the need to specify the flux at the plasma boundary, ψ_p, since this becomes part of the solution. ψ_p is given by the value of ψ on that ψ = constant surface which touches the limiter. It would be convenient if the geometrical constraint could be identified with the physical limiter, but as we shall see below, this is not always possible.

Convergence difficulties with Picard's method and the geometrical constraint alone [8,14] can be obviated by treating some parameter in the current density (the inhomogeneous term) as an eigenvalue. The most obvious method is to apply the total current constraint. This is also highly desirable from an experimental point of view as the plasma current is usually known precisely, particularly when the code is used as a diagnostic aid.

If we write our equations as $L\psi = \lambda G(\psi)$, then at each step of the iteration, we integrate the right hand side over the plasma (with appropriate weights if necessary). This gives the latest estimate of the total plasma current. λ is then adjusted to give the required total current and the iteration is then repeated until convergence is achieved.

Depending on whether one or both constraints are applied a number of rules have been discovered, which should ensure convergence.

(1) When only the geometrical constraint is applied, in axisymmetric geometry, in the form of a single point on $z = 0$, the point should be on the outside of the plasma.

(2) When both the geometrical (single point) and current constraints are applied, in axisymmetric geometry, the fixed point should be on the inside of the plasma.

(3) When both the geometrical and current constraints are applied in straight geometry or in low-β axisymmetric situations, a rail limiter (z = constant line) should be used for vertically elongated plasmas, and a point limiter for nearly circular plasmas.

Rules (1) and (2) are given by Cenacchi et al [2]. These are derived by a consideration of Shafranov's formula for the equilibrium of a circular plasma in a vertical field, [9]. Experience shows that for circular plasmas, faster convergence is achieved if (1) is used rather than (2). The first part of rule (3) is given by Thomas and Haas [14]. They show analytically that a uniform current density plasma in a straight quadrupole field, with

specified total current and width may have 2 height to width ratios according to the type of limiter. The method used in our codes can only give a height to width ratio of 1.84 with a point limiter. Use of the rail limiter gives a unique solution and essentially arbitrary height to width ratios may be computed. Computational experience [14] shows that this effect is present in axisymmetric geometry, at aspect ratios of 5:1 and higher. The second part of rule (3) also comes from a consideration of Shafranov's formula. It is easy to show that the use of a rail limiter with a circular plasma leads to a non convergent iteration.

2.3 Treatment of the Current Density

The next step in the procedure is the computation of the current density in the plasma. In straight geometry, the differential equation is

$$\begin{aligned} \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} &= - FF'-\mu_o p' \quad \text{in the plasma} \\ &= - \mu_o j_z \\ &= 0 \quad \text{in the vacuum} \end{aligned}$$

where $F(\equiv B_z)$ and p (the plasma pressure) are functions of ψ. In axisymmetric geometry the differential equation is

$$\begin{aligned} R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial\psi}{\partial R}\right) + \frac{\partial^2\psi}{\partial z^2} &= -FF' - \mu_o R^2 p' \quad \text{in the plasma} \\ &= - \mu_o Rj_\phi \\ &= 0 \quad \text{in the vacuum} \end{aligned}$$

where $F(\equiv RB_\phi)$ and p(the plasma pressure) are functions of ψ. [Throughout our calculations, SI units are used, where $\mu_o = 4\pi 10^{-7}$ henrys/metre]. The usual difference replacement for these equations is second order divided differences for the differential operator and a point value for the right hand side. Thomas [15] showed in a particular case that this treatment is not convergent as the mesh size tends to zero. It appears that the crude replacement of the plasma-vacuum interface is the cause of this behaviour. Consideration of the integral form of the differential equation reinforces this view. In straight geometry, the integral form is

$$\int_c \frac{\partial\psi}{\partial n}\, d\ell = - \mu_o \int_s j_z\, dS \qquad (1)$$

and in axisymmetric geometry, the form is

$$\int_c \frac{1}{R}\frac{\partial\psi}{\partial n}\, d\ell = - \mu_o \int_s j_\phi dS$$

In each case C and S are as in Fig. 1. The standard difference replacement of (1) would be

$$\frac{\psi_{ij+1} - \psi_{ij}}{H_N}\frac{1}{2}(H_E + H_W) + \frac{\psi_{ij-1} - \psi_{ij}}{H_S}\frac{1}{2}(H_E + H_W)$$
$$+ \frac{\psi_{i+1j} - \psi_{ij}}{H_E}\frac{1}{2}(H_N + H_S) + \frac{\psi_{i-1j} - \psi_{ij}}{H_W}\frac{1}{2}(H_N + H_S)$$
$$= - j_{z_{ij}}\frac{1}{4}(H_N + H_S)(H_E + H_W)\mu_o \quad . \qquad (2)$$

Note that the left hand side has the form obtainable by second order finite differences. Since this equation is unaltered if on the right side of (1) we take an exact approximation to the integral with j_z being constant in each mesh cell it is clear that when a mesh cell straddles the the interface, we are in fact replacing the interface by a step. In [15], the linear equations at each step were solved by SOR. Hence there was no restriction on the number of difference equations, (a restriction does arise in the method used in Section 2.5), and a thorough convergence test with a steadily decreasing mesh size could be performed. The results with difference replacement (2) are given in [15] and the lack of convergence is quite obvious.

The integral form of the equation suggests that we should perform the surface integral over the plasma alone. This makes the algorithm cumbersome since the shape of the interface has to be represented reasonably accurately. The loss in speed is acceptably small since most mesh cells lie completely in the vacuum or completely in the plasma and are not intersected by the interface. Hence the first step in the algorithm is to locate those mesh cells which straddle the interface. Since the solution ψ is smooth, even though the current density is possibly discontinuous we may accurately interpolate ψ at the corners of the mesh cells. These 4 values tell us whether the cell is completely in the vacuum, completely in the plasma or straddling the interface. In the vacuum case, the contribution to the right hand side of the difference equation is zero. In the second case, we do not have an interface to consider, but in order to retain uniformity of treatment we have used the following quadrature. From the values of ψ at the corners of the cell, we evaluate the current density at the 4 corners. The cell is then divided into 2 triangles by a diagonal line and the surface integral of the current density computed in each triangle using a linear interpolant as the integrand. The cell is divided into 2 triangles by a diagonal in the opposite direction and the integral computed. We then take the

mean of the 2 results. When the interface intersects a mesh cell we use the ψ values at the corners to interpolate the points where the interface crosses the cell sides. If the interface is allowed to cross each cell side at most once, then all the possible configurations are as shown in Fig.2. Thus we perform our integrations over either a pentagon, a quadrilateral or a triangle. The pentagon is split into 3 triangles in 5 possible ways. The mean of the 5 integrals is taken as the contribution to the right hand side. The quadrilateral is treated similarly to the rectangle and the triangle just has the one integral.

The considerable improvement in accuracy achieved by this scheme is demonstrated in the example in [15].

2.4 Computation of the Boundary Conditions

We cannot solve the set of difference equations without boundary conditions. From the properties of the vector potential it is easy to show that the boundary values are determined by the coil currents and the plasma currents. The plasma currents are given by the current density form and are in general unknown, a priori.

An elegant method for computing the boundary values has been given by Lackner [5]. By a simple application of Green's theorem he shows how the flux value at each boundary grid point is obtained by evaluating a line integral around the grid boundary. Assuming the boundary values are computed in only the upper half plane of a $N \times N$ grid, the number of subroutine calls is $8N^2$. This contrasts with the method described in [15] where the number of subroutine calls is $2\alpha N^3$, αN^2 being the number of grid points in the plasma $(\alpha < 1)$. As shown in Appendix I this gain in speed is at the expense of solving Poisson's equation once for each complete calculation of the boundary conditions. Following [16] a further improvement in speed is obtained by computing the boundary conditions at, say, every 5th step rather than at every step.

2.5 Solution of the Difference Equations

Since we always choose the grid to be a rectangle, we are able to take full advantage of the most efficient methods. The extra computation required to solve Poisson's equation turns out to be of the same order as that for computing the boundary conditions. In both the toroidal and the straight geometry codes the grid is chosen with an equal number of equispaced points in each direction. Thus in straight geometry, the local difference equation may be written

$$\frac{\Delta y^2}{\Delta x^2}\left[\psi_{i+1j} - 2\psi_{ij} + \psi_{i-1j}\right] + \psi_{ij+1} - 2\psi_{ij} + \psi_{ij-1} = \frac{-S_{ij}\, j_{z_{ij}}\, \Delta y}{\Delta x}$$

where $S_{ij} j_{z_{ij}}/\Delta x \Delta y$ is the contribution from the quadrature in section 2.3. Ordering the points along the rows, the set of difference equations is

$$\begin{bmatrix} A & I & & & & \\ I & A & I & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & I & A & I \\ & & & & I & A \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-2} \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} y_2 \\ y_3 \\ \cdot \\ \cdot \\ \cdot \\ y_{n-2} \\ y_{n-1} \end{bmatrix} \tag{3}$$

where A is the tridiagonal matrix,

$$\begin{bmatrix} -\lambda & 1 & & & & \\ 1 & -\lambda & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & 1 & -\lambda & 1 \\ & & & & 1 & -\lambda \end{bmatrix}$$

I is the identity matrix

and $\lambda = 2\left(1 + \frac{\Delta y^2}{\Delta x^2}\right)$. $x_k^T = (\psi_{2k}, \psi_{3k}, \ldots, \psi_{n-2k}, \psi_{n-1k})$ is the kth row of the solution ψ and y_k^T is a vector of terms involving $j_{z_{ij}}$ and the boundary conditions.

In axisymmetric geometry, the local difference equation may be written

$$\frac{\Delta z^2}{\Delta r^2}\left[\frac{\psi_{i+1j} - \psi_{ij}}{r_i + \frac{1}{2}\Delta r} + \frac{\psi_{i-1j} - \psi_{ij}}{r_i - \frac{1}{2}\Delta r}\right] + \frac{\psi_{ij+1} - 2\psi_{ij} + \psi_{ij-1}}{r_i} = -\, j_{\phi ij}\, S_{ij}\, \frac{\Delta z}{\Delta r}$$

where $S_{ij} j\phi_{ij}/\Delta r \Delta z$ is the contribution from the quadrature in section 2.3. Ordering the points along the rows, the set of difference equations is

$$\begin{bmatrix} B & D & & & & \\ D & B & D & & & \\ & \cdot & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot & \cdot \\ & & & D & B & D \\ & & & & D & B \end{bmatrix} \begin{bmatrix} w_2 \\ w_3 \\ \cdot \\ \cdot \\ w_{n-2} \\ w_{n-1} \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ \cdot \\ \cdot \\ z_{n-2} \\ z_{n-1} \end{bmatrix} \tag{4}$$

where B is the symmetric matrix

$$\begin{bmatrix} b_2 & c_2 & & & & \\ c_2 & b_3 & c_3 & & & \\ & \cdot & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot & \cdot \\ & & & c_{n-3} & b_{n-2} & c_{n-2} \\ & & & & c_{n-2} & b_{n-1} \end{bmatrix}$$

with $b_i = -\frac{2}{r_i} - \left(\frac{1}{r_i + \frac{1}{2}\Delta r} + \frac{1}{r_i - \frac{1}{2}\Delta r}\right)\frac{\Delta z^2}{\Delta r^2}$

and $c_i = \frac{\Delta z^2}{(r_i + \frac{1}{2}\Delta r)\Delta r^2}$

and D is the diagonal matrix

$$\begin{bmatrix} d_2 & & & & \\ & \cdot & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & d_{n-1} \end{bmatrix}$$

with $d_i = \frac{1}{r_i}$.

We may transform the system of equations (4) to the form (3) by setting

$$A = D^{-\frac{1}{2}} B D^{-\frac{1}{2}}$$

$$\underline{x}_i = D^{\frac{1}{2}} \underline{w}_i$$

and $$\underline{y}_i = D^{-\frac{1}{2}} \underline{z}_i \quad .$$

Hence in both straight and axisymmetric geometry, we have a set of equations

$$M\underline{x} = \underline{G}(\underline{x}) \tag{5}$$

to solve, where M has the form shown in (3). If we use Picard's method to solve (5), then the set of equations at each step may be solved using the Bunemann algorithm [17].

The minimum requirements for this algorithm are

(a) the boundary should be a rectangle,

(b) the number of points in one direction should be $2^n + 1$,

(c) the type of boundary condition on any side of the grid should not change on that side, although the boundary values are not restricted in any way,

(d) any variable coefficients or variable mesh should be independent of the direction referred to in (b).

In our implementation all these requirements are met and in addition the number of points in both directions is $2^n + 1$ and we do not have a variable mesh. Tridiagonal sets of equations have to be solved a large number of times and this is performed using the well known special case of Gaussian Elimination. We find a $65^2 \times 65^2$ matrix can be inverted on the ICL 4/70 in about 15 secs, which is some 10 times faster than optimal relaxation techniques.

2.6 Special Treatment of Shaped Cross Section Plasmas

During our calculations on the straight quadrupole problem (see Section 3), two further aspects of the codes had to be developed. As is well known, a plasma with a vertically elliptic cross section is unstable to a vertical shift if the height to width ratio of the plasma is greater than 1, [9] . This instability occurred in our equilibrium calculations for appreciable values of b/a when the plasma boundary was constrained to touch a rail limiter in the upper half plane. The plasma was seen to lift itself during the iterations until it vanished into a point on the limiter. We are able to prevent this behaviour if we symmetrise top and bottom. This has the effect of reflecting the limiter in the lower half plane, thus fixing the plasma boundary both top and bottom. This remedy has also been suggested by Cennacchi et al [2] .

In the equilibrium equation, the plasma and vacuum regions are usually distinguished by the value of ψ being either above or below the interface value. However for equilibria in which the separatrix lies close to the plasma boundary, it is possible for points in the vacuum region to have the same flux values as points within the plasma. Hence, examination of the ψ values is not sufficient to distinguish between plasma and vacuum. We are forced to introduce a partly geometric method of distinction. In the codes, we have required the user to specify a suitable rectangle with the field zeros of the separatrix lying outside it. This rectangle is used in the following way; if a point lies outside the rectangle, then it is defined to be in the vacuum region, independent of ψ, and if the point lies inside the rectangle then the usual definition of plasma - vacuum applies.

When positioning this so called 'virtual limiter' (introduced by Suzuki [3]), it is necessary to consider the form of vacuum field and the limiter (referred to as the 'numerical limiter') described in Section 2.2. For example, if the numerical limiter is a rail then a separatrix field zero cannot lie below the rail other than on $z = 0$. Hence the position of the rail defines the $z =$ constant sides of the virtual limiter. If there is a field zero on $z = 0$, it is not always possible to predict its position and in this case ' trial and error' positioning of the virtual limiter is necessary.

2.7 Summary of the Algorithm

These then are the principal features of the equilibrium calculation in both straight and axisymmetric geometry. A typical step in the iterative loop is as follows. We suppose that ψ^n has been computed.

(1) Apply the geometric constraint to obtain the new plasma shape. In the present version of the codes, the numerical limiter can be a rectangle, a rail on z (or y) = constant or a point on the plane of symmetry. The limiter is scanned for the minimum (assuming $I_p < 0$) value of ψ^n. This corresponds to the outermost closed surface to touch the limiter.

(2) Taking account of the virtual limiter, the current contributions to the right hand side of the difference equations are evaluated from the integral form.

(3) Symmetry about the mid-plane is applied using these contributions, the total plasma current I_p^n is computed, and assuming the current constraint is to be applied we appropriately adjust the scale factor in the current density. If the scale factor at step n is a^n and the required total current is I_p, we set

$$a^{n+1} = \frac{a^n}{I_p^n} I_p \quad .$$

(4) We examine the convergence of ψ_p^n and I_p^n. The user is allowed to specify the relative accuracies of these variables. The maximum residual is also evaluated as a guide. If the convergence criteria are satisfied, we go to step (8).

(5) If this is an iteration at which the boundary values are updated, we follow the procedure as given in Appendix I. The boundary values are symmetrised about the mid-plane.

(6) The right hand sides of equations (5) are constructed from the latest boundary values and the current contributions. The equations are then solved for ψ^{n+1} using the Bunemann algorithm.

(7) We return to step (1).

(8) Graphical output is produced together with various output variables such as β, β_I and q.

3. COMPARISONS WITH ANALYTIC AND NUMERICAL SOLUTIONS OF MULTIPOLE EQUILIBRIA

In this section we compare results produced by the codes with known analytic and numerical results. Essentially analytic results are restricted to the straight equilibrium analysed by Strauss [7] and the more approximate toroidal results of Mukhovatov and Shafranov [9]. As an example of known numerical results we consider some results obtained by Feneberg and Lackner [6] in a study of a belt pinch.

The mathematical equations of a multipole equilibrium are inherently nonlinear because of the presence of the free boundary. Consequently very little analysis has been possible. However Strauss [7] has produced an asymptotic solution for the equilibrium of a straight uniform current density plasma in a straight quadrupole field. Papaloizou et al [8] have extended this work, and this problem provides an ideal test case. Indeed, the analytic work gives a valuable insight into a number of convergence phenomena which have been observed in other codes, see [14].

3.1 Comparison with an Asymptotic Solution in Straight Geometry

The conductor configuration is illustrated in Fig.3. We choose the plasma current to be negative, so the cross section is elongated vertically and compressed horizontally. The analysis shows that provided the conductors are sufficiently far away, the plasma cross section is a perfect vertical ellipse. Strauss shows that the value of j, the current density determines whether, for a given external field there are no equilibria, one or two values of b/a, b and a being respectively the height and width of the plasma cross section. The two values coalesce into one value when b/a = 2.92, the bifurcation point. Furthermore since the value of j determines only the

height to width ratio b/a there is an infinite family of equilibria for each value of b/a.

In our calculations, using the straight geometry code, we must use a geometric constraint. We have constrained the plasma boundary to pass through the point $y = B$ in Fig.3, thus fixing the height of the plasma. Computations in which we fix the width of the plasma are described in [14] . The symmetry of the problem allows us to fix the plasma boundary on the y-axis. In general of course we would use a rail-limiter where the plasma is allowed to touch anywhere along the line $y = B$. We have taken the initial guess as an ellipse with $b/a = 2$ for all the runs which we report below. Tests with other initial guesses show that the choice of initial guess is not important.

We perform several series of computations. In each series, the positions of the wires, the currents in the wires and the limiter position are all fixed and the plasma shape is varied by adjusting the total plasma current. Following the analysis, we plot b/a against j, the current density, our results being shown as the discrete points in Fig.4. The solid curve is obtained from the analysis of Strauss [7]. His equation (23) can be written as

$$Kx^3 - x^2 + Kx + 1 = 0$$

which we solve numerically [18]. This equation has 2 positive roots x_1 and x_2 and one negative root. In our units

$$K = \frac{16\ 10^{-7}\ I_w}{jd^2}$$

where I_w and j have the same sign and b/a is given by

$$b/a = \frac{x + 1}{x - 1}$$

The analytic result is obtained by assuming that the wires are a large distance from the plasma. In practice, since feedback stabilisation is necessary for an elongated plasma, the wires must be placed close to the plasma. Hence our computations are performed initially with $B/d = 0.5$ to show the behaviour when the configuration is of practical interest, and subsequently with $B/d = 0.025$ to compare with the analytic results. Surprisingly, the value of b/a at the bifurcation point is insensitive to the positioning of the wires, although the curve shifts significantly to the right as the wires are brought closer to the plasma. Some discrepancy must be expected

because as the wires are brought in, the plasma shape will suffer distortion away from an ellipse. Calculations with $B/d = 0.68$ show a curve which is even further shifted to the right, but still the bifurcation point is quite close to the analytic value.

3.2 Comparison with Predictions of Shafranov's Formula

In the design of many tokamaks with circular cross section plasmas, considerable use has been made of Shafranov's equation [9]

$$B_v = \frac{\mu_o I_p}{4\pi R_p}\left[\ell n\,\frac{8R_p}{a} + \beta_I + \frac{\ell_i}{2} - \frac{3}{2}\right] \qquad (6)$$

In this equation, I_p is the plasma current, R_p is the distance of the geometric centre of the plasma from the axis of symmetry, a is the minor radius of the plasma (a circular plasma being assumed), β_I is the poloidal beta and ℓ_i is the self inductance of the plasma. The vacuum vertical field B_v is assumed to be uniform. In practice B_v will not be uniform across the plasma. Nevertheless if B_v is replaced by the vacuum vertical field at the centre of the plasma, equation (6) is satisfied to quite reasonable accuracy. This has been verified numerically by several authors [6,19] .

Using the TOSCA [10,11] configuration shown in Fig.5 we have computed equilibria with the limiter placed in several positions. A flat current ($\mu_o R j_\phi \sim A + BR^2$) has been used and taking $\ell_i = \frac{1}{2}$, the values given by the code have been substituted into the right hand side of equation (6). The resulting values of B_v are plotted against R_p, in Fig.6 in which we plot the actual vertical field, for comparison. Agreement is within 1% for all computed equilibria.

Incidentally, our results have been computed firstly by specifying the total plasma current, with the limiter on the inside and secondly by specifying the scale factor in the current density, with the limiter on the outside. The scale factor and limiter positions in the second set of runs were the values given by the first set. In each case the second method and the first method gave identical answers, strongly suggesting a unique solution.

Cenacchi et al [2] have shown that equation (6) admits the possibility of 2 solutions with the same limiter position and current density but different total currents. In the case of TOSCA parameters it is easy to show that this non-uniqueness is not possible.

3.3 Comparisons with Belt Pinch Equilibria

Feneberg and Lackner [6] studied a belt pinch configuration consisting of 2 vertical columns of 9 wires. They plotted plasma ellipticity against current density and found bifurcation, the bifurcation point being at $b/a \sim 2.9$. We have reproduced their results by fixing the plasma current and elongating the plasma by drawing apart a pair of rail limiters. However, in addition to the above bifurcation we find a further bifurcation at $b/a = 1.9$. This result is found by plotting b/a against a^2/I_p as suggested in [14], where a quadrupole external field is used. Asymptotic expansion of the belt pinch stream function in straight geometry shows that it is indeed essentially a quadrupole.

We have studied 2 further straight configurations in which a vertically elongated uniform current density plasma is produced. The first configuration consists of 2 wires on the $Y = 0$ plane carrying equal currents which are in the opposite direction to the plasma current. The second configuration is similar to a belt pinch and consists of two columns of 3 wires, with all their currents equal but in the opposite direction to the plasma current. The first configuration produces a race track shape plasma cross section and the second produces a doublet-like shape cross section. Nevertheless if we take a as the maximum width of the plasma, we find that both configurations have a bifurcation point at $b/a \sim 2.9$. These results strongly suggest that when a symmetrically elongated flat current plasma is produced, the bifurcation point is at $b/a \sim 2.9$ - the vacuum field does not have to be a quadrupole. [We define a symmetric plasma to be symmetrically placed with respect to the conductors and to have negligible triangularity.]

4. USE OF THE CODE AS A DIAGNOSTIC AID IN TOSCA

In present experiments, it is rather difficult to measure either the toroidal current density or the plasma shape. It is current practice [20,21,22] to use a discrete conductors MHD equilibrium code as a diagnostic aid for studying these parameters. The data for these codes are the experimental parameters for which accurate measurements are available. Typically these parameters will be the total plasma current, the currents and positions of the external conductors, a point on the plasma boundary (if the plasma is known to touch the limiter) and β_I. The accuracy of the computed equilibrium is given by comparing the fields measured experimentally and those computed in the code. Poor accuracy implies that the current density profile used in the code is not correct.

A complete description of the use of the code as a diagnostic aid for TOSCA is given in [10,11] .

ACKNOWLEDGEMENTS

Stimulating conversations with K W Morton, P J Fielding, A Wootton and R C Grimm have been of considerable help in this work. I should also like to express my thanks to F A Haas for his invaluable encouragement and advice.

REFERENCES

[1] THOMAS, C.Ll. and HAAS, F.A., Computation of MHD Equilibria in Tokamak, Culham Report CLM-R133, 1974.

[2] CENACCHI, G., GALVÃO, R. and TARONI, A., Numerical Computation of Axisymmetric MHD Equilibria Without Conducting Shell, Nuclear Fusion, 16, p.457, 1976.

[3] SUZUKI, Y., Free Boundary MHD Equilibria in Axisymmetric Tori, Nuclear Fusion, 14, p.345, 1974.

[4] CHU, M.S., DOBROTT, D., JENSEN, T.H. and TAMANO, T., Axially Symmetric Magnetohydrodynamic Equilibria with Free Boundaries and Arbitrary Cross Section, Physics of Fluids, 17, p.1183, 1974.

[5] LACKNER, K., Computation of Ideal MHD Equilibria, Computer Physics Communications, 12, p.33, 1976.

[6] FENEBERG, W. and LACKNER, K., Multipole Tokamak Equilibria, Nuclear Fusion, 13, p.549, 1973.

[7] STRAUSS, H.R., Bifurcation of Elliptic Equilibrium, Physics of Fluids, 17, p.1040, 1974.

[8] PAPALOIZOU, J.C.B., REBELO, I., FIELD, J.J., THOMAS, C.Ll. and HAAS, F.A., Magnetohydrodynamic Equilibria in a Straight Magnetic Quadrupole Field, Nuclear Fusion, 17, p.33, 1977.

[9] MUKHOVATOV, V.S. and SHAFRANOV, V.D., Plasma Equilibrium in a Tokamak, Nuclear Fusion, 11, p.605, 1971.

[10] CIMA, G., ROBINSON, D.C., THOMAS, C.Ll. and WOOTTON, A.J., Shaping and Compression Experiments in a Tokamak, Plasma Physics and Controlled Nuclear Fusion Research, 1976 , International Atomic Energy Agency, Vienna, 1977.

[11] WOOTTON, A.J. and ROBINSON, D.C., Equilibrium, Stability and Confinement of Elliptic Cross Sectioned Tokamak Plasmas, 8th European Conference on Controlled Fusion and Plasma Physics, 1, p.42, 1977.

[12] CALLEN, J.D. and DORY, R.A., Magnetohydrodynamic Equilibria in Sharply Curved Axisymmetric Devices, Physics of Fluids, 15, p.1523, 1972.

[13] CHANCE, M.S., DEWAR, R.L., GLASSER, A.H., GREENE, J.M., GRIMM, R.C., JARDIN, S.C, JOHNSON, J.L., ROSEN, B., SHEFFIELD, G.V. and WEIMER, K.E. Study of Magnetohydrodynamic Modes in Tokamak Configurations with Noncircular Cross Sections, Plasma Physics and Controlled Nuclear Fusion Research, 1974, International Atomic Energy Agency,Vienna, 1975.

[14] THOMAS, C.Ll and HAAS, F.A., On the Bifurcation of Magnetohydrodynamic Equilibria in a Magnetic Quadrupole Field, Culham Report CLM-P540, 1978, (to appear in Nuclear Fusion).

[15] THOMAS, C.Ll., The Numerical Calculation of Axisymmetric MHD Equilibria in Discrete Conductor Configurations, Computational Methods in Classical and Quantum Physics (Ed. M.B. Hooper), Transcripter Books, 1977.

[16] JOHNSON, J.L., Private Communication.

[17] BUZBEE, B.L., GOLUB, G.H. and NIELSON, C.W., On Direct Methods for Solving Poisson's Equations, SIAM J. Num. Anal., 7, p.627, 1970.

[18] MARTIN, T.J., Root Finding and the Solution of Dispersion Equations, Culham Report CLM-PDN 3/71, 1971.

[19] SAKURAI, K., TANAKA, Y. and OKUDA, T., Equilibrium in a Multipole Tokamak, J. Phys. Soc. of Japan, 37, p.1108, 1974.

[20] FREEMAN, R.L., ADCOCK, S.J., BAUR, J.F., BROOKS, N.H., DeBOO, J.C., FISHER, R.K., GUSS, W.C., HELTON, F.J., HSIEH, C.L., JENSEN, T.H., LIETZKE, A.F., LOHR, J.M., MAHDAVI, M.H., MATSUDA, K., MOELLER, C.P., OHKAWA, T., OHYABU, N., PRAGER, S.C., RAWLS, J.M., TAMANO, T., VANEK, V. and WANG, T.S., Doublet II A Experiments, Plasma Physics and Controlled Nuclear Fusion Research, 1976, International Atomic Energy Agency, Vienna, 1977.

[21] TOYAMA, H., INOUE, S., ITOH, K., IWAHASHI, A., KANEKO, H., MAKISHIMA, K., OCHIAI, I., SHINOHARA, S., SUZUKI, Y. and YOSHIKAWA, S., Experiments on Non Circular Tokamak and Related Topics, Plasma Physics and Controlled Nuclear Fusion Research 1976, International Atomic Energy Agency, Vienna, 1977.

[22] GRAFFMANN, E., HOENEN, F., KALECK, A., KÖNEN, L. and SCHLÜTER, J., Energy Confinement and Beta Limit in the Belt Pinch Experiment TENO, Plasma Physics and Controlled Nuclear Fusion Research, 1976, International Atomic Energy Agency, Vienna, 1977.

[23] FIELDING, P.J. and BEVIR, M.K., The Numerical Solution of Dirichlet Problems with Axisymmetry by an Integral Equation Method, Culham Report CLM-R146, 1975.

[24] WOOTTON, A.J., Private Communication.

[25] HAAS, F.A., Sharp Boundary Model of a High Pressure Tokamak, Physics of Fluids, 15, p.141, 1972.

APPENDIX I

IMPLEMENTATION OF LACKNER'S METHOD FOR COMPUTING THE BOUNDARY CONDITIONS [5]

In straight geometry, the method derives from a consideration of Green's theorem in (x,y) coordinates

$$\int_R (u\ \nabla^2 v - v\ \nabla^2 u)dS = \oint_{\partial R} \left(u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n}\right) d\ell \tag{A1}$$

We set u equal to the solution $\hat{\psi}$ of

$$\begin{aligned} \nabla^2\hat{\psi} &= -\ \mu_o j_z \qquad \text{in} \quad R \\ \hat{\psi} &= 0 \qquad\qquad \text{on} \quad \partial R \end{aligned} \tag{A2}$$

where R is the interior of the grid rectangle and ∂R is its boundary. We set v equal to the flux function G of a filament infinite in the z direction, $(G = -\ \frac{\mu_o}{2\pi}\ \ell n[(x - x_c)^2 + (y - y_c)^2]^{\frac{1}{2}})$. Now since $\nabla^2 v = -\ \delta(x - x_c, y - y_c)$, where (x_c, y_c) are the coordinates of the filament, we have

$$-\iint_R G\nabla^2\hat{\psi}\ dxdy = -\oint_{\partial R} G \frac{\partial\hat{\psi}}{\partial n}\ d\ell + \hat{\psi}(x_c, y_c)$$

i.e.

$$-\iint_R G\mu_o j_z\ dxdy = -\oint_{\partial R} G \frac{\partial\hat{\psi}}{\partial n}\ d\ell + \hat{\psi}(x_c, y_c) \tag{A3}$$

therefore

$$\psi(x_c, y_c) = +\oint_{\partial R} \frac{\mu_o}{2\pi}\ \ell n[(x - x_c)^2 + (y - y_c)^2]^{\frac{1}{2}} \frac{\partial\hat{\psi}}{\partial n}\ d\ell$$

To get the total value of ψ on the boundary we must add in the contribution from the external conductors. Computationally we compute $\hat{\psi}$ by applying the Bunemann algorithm. However, the computation of $\psi(x_c, y_c)$ is complicated by the singularity in $\ell n[(x - x_c)^2 + (y - y_c)^2]^{\frac{1}{2}}$ at (x_c, y_c). Fortunately we are in a position to subtract the singularity. If we write the integral in (A3) as

$$\int G(x,y,x_c,y_c)\left(\frac{\partial\hat{\psi}}{\partial n}(x,y) - \frac{\partial\hat{\psi}}{\partial n}(x_c,y_c)\right)d\ell + \frac{\partial\hat{\psi}}{\partial n}(x_c,y_c)\int G(x,y,x_c y_c)\,d\ell \tag{A4}$$

the application of a quadrature rule to the first integral will have zero contribution at (x_c,y_c). Because of the form of $G(x,y,x_c,y_c)$ the second integral can be evaluated analytically. We have, in the notation of Fig.7

$$\int_{x_L}^{x_R} \ln[x - x_c)^2 + (y - y_c)^2]^{\frac{1}{2}}dx$$

$$= \lim_{\varepsilon\to 0}\left[\int_{x_L}^{x_c-\varepsilon} \ln(x_c - x)dx + \int_{x_c+\varepsilon}^{x_R} \ln(x - x_c)dx\right]$$

$$= (x_c - x_L)\ln(x_c - x_L) + (x_R - x_c)\ln(x_R - x_c) - (x_R - x_L)$$

The integral in (A3) is performed around all 4 sides of the mesh rectangle, but we only apply (A4) on the side containing the flux evaluation point. Quadratures are performed by Simpson's rule with the trapezoidal rule applied in the interval containing the singularity. The combination of rules is necessary because the singularity splits the range of integration into parts which may not have odd numbers of points which is a requirement for Simpson's rule.

In axisymmetric geometry, the boundary value equation is derived in an analogous fashion. We have

$$\psi(r_c,z_c) = \hat{\psi}(r_c,z_c) - \oint_{\partial R} \frac{G(r_c,z_c,r^*,z^*)}{r^*}\frac{\partial\hat{\psi}}{\partial n}(r^*,z^*)d\ell^* \tag{A5}$$

where $G(r_c,z_c,r^*,z^*) = \dfrac{\mu_o}{\pi}\left(\dfrac{r_c r^*}{k^2}\right)^{\frac{1}{2}}\left[\left(1 - \dfrac{k^2}{2}\right)K(k) - E(k)\right]$

and $$k^2 = \frac{4r_c r^*}{(r_c + r^*)^2 + (z_c - z^*)^2}$$

Following Fielding and Bevir [23], the singularity is subtracted by rewriting the integral in (A5) as

$$- \oint_{\partial R} \left[\frac{G(r_c, z_c, r^*, z^*)}{r^*} \frac{\partial \hat{\psi}}{\partial n} (r^*, z^*) - \frac{1}{r_c \pi} \left\{ \frac{1}{2} \ln \left[\frac{8r_c}{[(r_c - r^*)^2 + (z_c - z^*)^2]^{\frac{1}{2}}} \right] - 1 \right\} \frac{\partial \hat{\psi}}{\partial n} (r_c, z_c) \right] ds^*$$

$$- \oint_{\partial R} \frac{1}{r_c \pi} \left\{ \frac{1}{2} \ln \left[\frac{8r_c}{[(r_c - r^*)^2 + (z_c - z^*)^2]^{\frac{1}{2}}} \right] - 1 \right\} \frac{\partial \hat{\psi}}{\partial n} (r_c, z_c) ds^* \qquad \text{(A6)}$$

The second integral in (A6) may be evaluated analytically. When a quadrature rule is applied to the first integral in (A6), the contribution from the point (r_c, z_c) is zero.

APPENDIX II

INPUT AND OUTPUT FACILITIES OF THE AXISYMMETRIC AND STRAIGHT GEOMETRY CODES FOR COMPUTING MULTIPOLE MHD EQUILIBRIA

Provided the user is willing to study current densities which are at most linear in ψ, each code is in an executable form and only requires a data file. The data required to run each code are basically the same, although the axisymmetric version has rather more facilities since it will clearly be used much more than the straight version. Data input is in the form of a set of NAMELISTs, the data being separated into several natural groupings. We begin by listing the data for the axisymmetric code, with explanations for each data item.

(1) Namelist NRSTRT:

IRSTRT = 0 if no restart is required.
= 1 if restart is required.
= 2 if the initial guess for ψ is to be taken from a previous run

The restart facility allows the user to perform several short runs instead of one long run.

(2) Namelist NIO:

IGRAF = 0 if graphical output is not required.
= 1 if graphical output is required.

NCONT is the number of contour heights (flux values) which are required. The number of heights must be less than 65 and the default value is 21. The value of q, the safety factor, is calculated on each surface.

ISTORE = 1 if data is to be written to a file for a subsequent restart
= 0 if a subsequent restart is not required.

The default value is zero.

(3) Namelist NGEOM:

RMINOR is the half width of the grid rectangle in the R (horizontal) direction, measured in metres.

RMAJOR is the distance in metres from the axis of symmetry to the centre of the grid rectangle. Hence the left hand side of the rectangle is a distance RMAJOR-RMINOR from the axis of symmetry and the right hand side is a distance RMAJOR+RMINOR from the

axis of symmetry.

HEIGHT is the half height in the z direction (vertical) of the grid rectangle, measured in metres.

INDEX determines the number of grid points in both directions. The actual number of points in each direction is $2^{\mathrm{INDEX}} + 1$, where INDEX $\leqslant 6$.

(4) Namelist NPARAM:

A,B,C,D,E, are the coefficients in the right hand side of the MHD equilibrium equation. We have taken $\mu_o R j_\phi = A\{[B + C(\psi - \psi_p)] + \mu_o R^2[D + E(\psi - \psi_p)]\}$. A certain amount of care is necessary when choosing these coefficients. In the code the sign convention is $I_p > 0 \Rightarrow \psi > \psi_p$ in the plasma, where ψ_p is the flux at the plasma boundary, and $I_p < 0 \Rightarrow \psi < \psi_p$ in the plasma. If we assume that reversed currents are not present (although in principle the code will treat this situation), then A,B,C,D,E must satisfy this convention. Hence if $I_p < 0$ then $A < 0$, B and D $\geqslant 0$ and C and E $\leqslant 0$ satisfies the convention. B,C,D and E are under user control. A useful guide for selecting these parameters has been given by Wootton [24]. He has considered the flat current circular equilibrium studied by Haas [25], and has shown that for given I_p, β_I, minor radius ,a, and major radius R_o, the parameters should satisfy

$$AD = \frac{I_p\ \beta_I}{\pi a^2 R_o}$$

$$AB = \frac{\mu_o I_p R_o (1 - \beta_I)}{\pi a^2}$$

A is adjusted by the code if the plasma current I_p is to be constrained, but otherwise A retains the value specified.

IPSIP1,IPSIP2,JPSIP1 and JPSIP2 determine the limiter. For most applications either a rail or point limiter is appropriate. A rail should be used for a vertically elongated plasma. In this case set IPSIP1 = 2 and IPSIP2 = 2^{INDEX}. JPSIP1 and JPSIP2 are determined by z_{LIM}, the z-coordinate of the rail where JPSIP1= JPSIP2 = $2^{\mathrm{INDEX}-1} + 1 + z_{LIM}/\Delta z$. and $\Delta z = \dfrac{2 \times \mathrm{HEIGHT}}{2^{\mathrm{INDEX}}}$. Since symmetry about z = 0 is applied, JPSIP1 and JPSIP2 take the same value. For

a circular plasma, a point limiter on $z = 0$ should be used. The convergence properties of a point limiter calculation have already been discussed. Suppose R_{LIM} is the R-coordinate of the limiter. Then, set JPSIP1 = JPSIP2 = $2^{\text{INDEX}-1} + 1$ and IPSIP1 = IPSIP2 = $1 + (R_{LIM} - \text{RINNER})/\Delta R$ where $\Delta R = \dfrac{2 \times \text{RMINOR}}{2^{\text{INDEX}}}$ and RINNER = RMAJOR-RMINOR.

TORCUR is the plasma current measured in amperes. If the total current constraint is not to be used, this variable may be omitted from the data.

ILIM1,ILIM2,JLIM1 and JLIM2 determine the limiting box which is necessary when a separatrix is present. If no separatrix is expected, set ILIM1 < 1, JLIM1 < 1, ILIM2 > $2^{\text{INDEX}} + 1$ and JLIM2 > $2^{\text{INDEX}} + 1$. This tells the code that within the grid rectangle, the normal definitions of plasma and vacuum apply. Otherwise suppose the separatrix field zeros are expected to lie outside a rectangle defined by $R_{s1} \leq R \leq R_{s2}$, $z_{s1} \leq z \leq z_{s2}$, where $z_{s1} = - z_{s2}$. Then set ILIM1 = $1 + (R_{s1}-\text{RINNER})/\Delta R$, ILIM2 = $1 + (R_{s2}-\text{RINNER})/\Delta R$, JLIM1 = $1 + \dfrac{(z_{s1}+\text{HEIGHT})}{\Delta z}$ and JLIM2=$1 + \dfrac{(z_{s2} + \text{HEIGHT})}{\Delta z}$

A certain amount of 'trial and error' is necessary when placing the box. The simple rule is that if the box allows currents within the grid rectangle, other than in the plasma, the calculation should be repeated with an appropriate adjustment to the box.

BZERO is the vacuum toroidal field at R=RMAJOR, measured in tesla (1 tesla = 10kG). This value is used in the calculation of q and β where β is the ratio of total plasma pressure to total magnetic energy within the plasma.

(5) Namelist NSOLVE:

NSTEPS is the total number of iterative steps to be performed, unless the convergence criterion determined by EPS below is satisfied first. This number is problem dependent and its default value is 50.

NOIT is the interval between computation of the boundary values, i.e. the boundary values are up-dated every NOIT steps. The default value is 5.

EPS is the convergence criterion. The iteration will stop when $|I_{p_n}^{n+1} - I_p^n| \leq \varepsilon |I_p^{n+1} + I_p^n|$ and $|\psi_p^{n+1} - \psi_p^n| \leq \varepsilon |\psi_p^{n+1} + \psi_p^n|$, where I_p^n and ψ_p^n are the plasma current and flux on the

plasma boundary, respectively, at the nth iteration. The default value is 10^{-5}.

ICURNT = 0 if the total current constraint is not used.

= 1 if the total current constraint is used.

(6) Namelist NSETUP:

XP,YP,RP,HP are coefficients in the form of the initial guess. This form is under user control, see Appendix III, but the standard version of the code assumes a circular shape for the plasma. In this case, the equation of the flux surfaces is

$$\psi(R,Z) = HP[RP - \sqrt{[(Z - XP)^2 + (R - XP)^2]}]$$

where R is measured from R = RMAJOR. Note that HP must have the same sign as I_p. The position of the limiter must be taken into account when ensuring that the initial plasma shape lies completely within the grid rectangle. Note that by setting IRSTRT=2, a previous solution may be used as the initial guess.

(7) Namelist NCOIL:

NCOILS = the number of filamentary windings (≤ 50)

RCOIL is the array of R-coordinates of the windings, measured in metres from the axis of symmetry

ZCOIL is the array of Z-coordinates of the windings, measured in metres.

CCOIL is the array of currents in the windings, measured in amperes.

(8) Namelist NPROUT:

NPRINT is the number of positions at which fields are to be calculated. This number (≤ 100) may be zero.

IRZ = 1 if the cylindrical components B_R and B_Z are to be computed at the requested positions.

= 0 if the polar components B_r and B_θ are to be computed at the requested positions. In this co-ordinate system, r is measured from the point R=RMAJOR, Z = 0.

RPRINT is the array of R coordinates of the points at which fields are to be calculated. R is measured in metres from the axis of symmetry.

ZPRINT is the array of Z-coordinates of the points at which fields are to be calculated. Z is measured in metres.

As an example we consider the data set used for the calculation of a DITE equilibrium. The geometry and flux surfaces are shown in Fig.8.

```
&NRSTRT
IRSTRT=0
&END
&NIO
IGRAF=1,NCONT=21
&END
&NGEOM
RMINOR=0.33,HEIGHT=0.23,RMAJOR=1.27,INDEX=5
&END
&NPARAM
A=1.0,B=0.0,C=1.256637D-6,D=0.0,E=1.0,
IPSIP1=2,IPSIP2=2,JPSIP1=17,JPSIP2=17,
ILIM1=-100,ILIM2=100,JLIM1=-100,JLIM2=100,
TORCUR=-50.0D3,BZERO=1.0
&END
&NSOLVE
NSTEPS=50,NOIT=5,EPS=1.0D-5,ICURNT=1
&END
&NSETUP
XP=1.0,YP=1.0,RP=-1.0,HP=0.0
&END
&NCOIL
NCOILS=11,
RCOIL=1.91,1.91,1.53,1.53,1.582,1.582,0.903,0.903,0.85,0.85,
.201,
ZCOIL=0.87,-0.87,0.37,-0.37,0.324,-0.324,0.277,-0.277,
0.19,-0.19,0.0,
CCOIL=20.0D3,20.0D3,3.5D3,3.5D3,-3.5D3,-3.5D3,-3.5D3,-3.5D3,
3.5D3,3.5D3,-839.0D3
&END
&NPROUT
NPRINT=0,IRZ=1,
&END
```

No restart facilities are required so ISTORE takes its default value of 0. The current density form is

$$\mu_o R j_\phi = A[1.256637\ 10^{-6}(\psi - \psi_p) + \mu_o R^2(\psi - \psi_p)] \quad .$$

Since $\beta_I \sim 0.5$, the coefficients have been chosen to satisfy the approximate relation $\beta_I = \frac{\mu_o E}{C + \mu_o E}$, see Haas [25]. No separatrix is expected within the grid rectangle and a point limiter on the inside of the plasma is used, since the total current constraint is used. The value of the vacuum toroidal field is 1 tesla, 127 cm from the axis of symmetry. 11 poloidal field windings are used. The first 10 are the actual windings in the apparatus and the last winding simulates the vertical field due to the presence of the iron core. No print out of the fields is required.

Output facilities are both graphical and printed. Graphical output

consists of the flux surfaces. The plasma boundary is plotted as a broken contour. Each surface has an identifying number plotted at some point along it. This number corresponds to the number in the printed output. Other printed output consists of some details of the iteration, the flux values along the R and Z coordinates of the magnetic axis,

$$\beta\left(= \frac{2\mu_o \int p\,dS}{\int B_\phi^2 dS}\right) \quad \text{and} \quad \beta_I\left(= \frac{8\pi}{\mu_o}\,\frac{\int p\,dS}{I_p^2}\right) \quad .$$

The data required by the straight geometry code is essentially a subset of the above data. Namelists are used in a similar fashion.

(1) Namelist NIO:

IGRAF = 1 if graphical output in the form of flux surface plots is required.

= 0 if graphical output is not required.

NCONT = the number of contour heights (flux values). The default value is 21.

(2) Namelist NGEOM:

XMINOR is the half-width of the grid rectangle in the X direction, measure in metres.

YMINOR is the half-width of the grid rectangle in the Y direction, measured in metres. The origin of the co-ordinates is at the centre of the rectangle.

INDEX gives the number of grid points in both directions. The number is $2^{\text{INDEX}} + 1$ and INDEX $\leq$ 6.

(3) Namelist NPARAM:

A,B and C are the coefficients in the right hand side of the MHD equilibrium equation. We have taken

$$\mu_o j_z = A[B + C(\psi - \psi_p)] \quad .$$

We always assume the plasma current to be negative and $\psi < \psi_p$ in the plasma. Hence a consistent choice of signs for A,B and C is $A < 0$, $B \geq 0$, $C \leq 0$. B and C retain values as specified by the user. A is adjusted internally if the total current constraint is applied, but otherwise retains its specified value.

IPSIP1,IPSIP2,JPSIP1, and JPSIP2 determine the limiter. Typically a rail or point limiter should be used. A rail limiter is

appropriate for a vertically elongated plasma. In this case set IPSIP1=2 and IPSIP2=2^{INDEX}. JPSIP1 and JPSIP2 are determined by the y-coordinate of the rail, Y_{LIM}. JPSIP2 should be set to $2^{\mathrm{INDEX-1}} +1 + Y_{\mathrm{LIM}}/\Delta Y$ where $\Delta Y = \frac{2 \times \mathrm{YMINOR}}{2^{\mathrm{INDEX}}}$ and JPSIP1 should be set equal to JPSIP2. Since symmetry about Y = 0 is applied, JPSIP1 and JPSIP2 take the same value. For a circular plasma, a point limiter on Y = 0 should be used. Suppose X_{LIM} is the X-coordinate of the limiter. Then set JPSIP1 = JPSIP2 = $2^{\mathrm{INDEX-1}} + 1$ and IPSIP1 = IPSIP2 = $1 + (X_{\mathrm{LIM}} + \mathrm{XMINOR})/\Delta X$ where $\Delta X = 2 \times \mathrm{XMINOR}/2^{\mathrm{INDEX}}$. Experience shows that, unlike the axisymmetric problem, there are no restrictions on the positioning of the point limiter.

CURENT is the total plasma current measured in ampères. If the total current constraint is not to be used, this variable may be omitted from the data.

ILIM1,ILIM2,JLIM1 and JLIM2 determine the limiting box which is necessary when a separatrix is present. If no separatrix is expected, set ILIM1 < 1, JLIM1 < 1, ILIM2 > $2^{\mathrm{INDEX}} + 1$ and JLIM2 > $2^{\mathrm{INDEX}} + 1$. Otherwise suppose the field zeros are expected to lie outside a rectangle defined by $X_{s1} \leq X \leq X_{s2}$, $Y_{s1} \leq Y \leq Y_{s2}$, where $Y_{s1} = - Y_{s2}$. Then set
ILIM1 = $1 + (X_{s1} + \mathrm{XMINOR})/\Delta X$, ILIM2 = $1 + (X_{s2} + \mathrm{XMINOR})/\Delta X$,
JLIM1 = $1 + (Y_{s1} + \mathrm{YMINOR})/\Delta Y$, and JLIM2 = $1+ (Y_{s2} + \mathrm{YMINOR})/\Delta Y$.
Again a certain amount of 'trial and error' may be necessary.

(4) Namelist NSOLVE:

NSTEPS is the maximum number of iterative steps to be performed. The default option is 50.

NOIT is the interval between computations of the boundary values. The default option is 5.

EPS is the convergence criterion. The iterations will stop when $|I_p^{n+1} - I_p^n| \leq \varepsilon\, |I_p^{n+1} + I_p^n|$ and $|\psi_p^{n+1} - \psi_p^n| \leq \varepsilon\, |\psi_p^{n+1} + \psi_p^n|$ where I_p^n and ψ_p^n are as defined in the axisymmetric data. The default option is 10^{-5}.

ICURNT = 0 if the total current constraint is not used
= 1 if the total current constraint is used.

(5) Namelist NSETUP:

XP,YP,RP,HP are coefficients in the form of the initial guess. The standard version of the code assumes an initially circular

plasma. The equation of the flux surfaces is

$$\psi(X,Y) = HP[RP - \sqrt{(X - XP)^2 + (Y - YP)^2}]$$

Note that HP must be negative and that the position of the limiter must be taken into account when defining the initial guess.

(6) Namelist NCOIL:

NCOILS = the number of filamentary windings ($\leqslant$ 32)

XCOIL is the array of X-coordinates of the windings, measured in metres.

YCOIL is the array of Y-coordinates of the windings, measured in metres.

CCOIL is the array of currents in the windings, measured in ampères.

As an example we consider a data file used in the uniform current density quadrupole calculations of Section 3.

```
&NIO
IGRAF=1,NCONT=21
&END
&NGEOM
XMINOR=0.1,YMINOR=0.1,INDEX=5
&END
&NPARAM
A=-0.011,B=1.0,C=0.0,D=0.0,E=0.0,
IPSIP1=17,IPSIP2=17,JPSIP1=25,JPSIP2=25,
ILIM1=2,ILIM2=32,JLIM1=9,JLIM2=25,
CURENT=-8.7D3
&END
&NSOLVE
NSTEPS=100,NOIT=5,EPS=1.0D-5,IPOINT=1,ICURNT=1
&END
&NSETUP
XP=0.75,YP=1.5,RP=-1.0,
HP=0.0
&END
&NCOIL
NCOILS=4,
YCOIL=0.11,0.0,-0.11,0.0,
XCOIL=0.0,-0.11,0.0,0.11,
CCOIL=-10.0D3,10.0D3,-10.0D3,10.0D3
&END
```

The half-width of the grid is 10 cm in both directions. Since the configuration is symmetric about both axes, the rail limiter reduces to a point on X = 0. The initial guess is not taken as a circle but an alternative subroutine which generates an ellipse is used. The form of the surfaces is $\psi(X,Y) = RP[1 - \frac{X^2}{XP^2} - \frac{Y^2}{YP^2}]$. The external windings are placed only 1 cm outside the grid.

APPENDIX III

ICL 4/70 RUNNING INSTRUCTIONS FOR THE CODES

In this Appendix we give listings of the files necessary to compose and execute the codes. We also give listings of the special function routines provided as standard. As in Appendix II, we begin with the axisymmetric code.

To compare and execute the axisymmetric code, the following line file should be run.

```
    // GROUP TEMP
    // SCHEDULE S/TRIALS,10,27
    // CONFG RSP=2E12,STORE=170
    // EXEC TROUT
    // TRIALS
    // COMPOSE TEMP.TOPEOL,REMJOB/TOPEOL.RUN1
    **OPTION TREE,ROOT
    **INCLUDE CLTTHE:TOPEOL.MAIN,CURENT,TRINGL,DATAD,BPHI
    **SUBST LEMCAL:MAGLIB.SUBS
    **SEGMENT 01,A
    **INCLUDE CLTTHE:TOPEOL.START1,RHS,SCRAPE,CONSTR,ANAL
    **INCLUDE CLTTHE:TOPEOL.INITAL,CNVERG,TRISOL,DIRECT,NPSIB
    **INCLUDE CLTTHE:TOPEOL.DATAIN,STORE,BPCALC,CONTOR,PLOTAF
    **INCLUDE CLTTHE:TOPEOL.BTHETA,CONTRA
    **SUBST CLTTHE:TOPEOL.GHOST
    **SEGMENT 02,A
    **INCLUDE CLTTHE:TOPEOL.P,BETA,ARINT,BPHISQ,FIELDS
    // ENDTRIALS
```

The execution file TOPEOL.RUN1(S) is

```
    // GROUP TEMP
    // SCHEDULE TOPEOL,200,6
    // CONFG RSP=2G09,STORE=575
    // FILE DSET97,RA,TOPEOL.DSET97(S0010)
    // FILE GRIDFL,RA,TEMP.HXPOLE(N0010)
    // FILE DSET70,RA,TOPEOL.DSET80(Z0010),VOL159,T25
    // FILE DSET80,RA,TOPEOL.DSET80(Z0020),VOL159,T25
    // FILE DSET99,RA,TEMP.DSET99(S0050)
    // EXEC TEMP.DSET99(S0050)
```

The meanings of these statements will be obvious to the experienced user butit is worth pointing out that the DSET70 file has been obtained as the DSET80 of a previous run.

Running times are configuration dependent, but 150 etus is typical for a plasma occupying 25% of the grid.

In the composition file, the user may if he wishes substitute his own files for the following routines

```
       DOUBLE PRECISION FUNCTION START(X,Y)
          IMPLICIT REAL*8 (A-H,O-Z)
          COMMON/SETUP/IRSTRT,AVERT,BVERT,ALPHA,BVAL,XP,YP,RP,HP
          START=RP*(1.0-X/XP*X/XP-Y*Y/YP/YP)+RP
       RETURN
       END
       FUNCTION RHS(RZEROS,PSI,R,Z)
          IMPLICIT REAL*8 (A-H,O-Z)
          COMMON/GEOM/NZP,NRP,ISYM,NHZ,NHR,INDEX,NGRID,NRED,NEQU
,NGM1,
     1    RMINOR,RMAJOR,HZ,HR,HEIGHT,RINNER,ROUTER,DELTA,DELTA1
          COMMON/PARAM/IPSIP1,IPSIP2,JPSIP1,JPSIP2,ILIM1,ILIM2,J
LIM1,
     1    JLIM2,A,B,C,PSIP,TORCUR,CALCUR,D,E,PIMU0,BZERO
          COMMON/SOLVE/NSTEPS,NOIT,ITNO,ICONVS,NUPDAT,ICONVP,IPO
INT,
     1    ICURNT,OMEGA,EPS,PHI(65,65),PHINEW(65,65),PHIVI1(65),
     2    PHIVI2(65),PHIVJ1(65),PHIVJ2(65),CE(65),CW(65),
     3    CTRANS(65),CZ,CZ2,OMEGA1,SOURCE(65,65),PHITWO(65,65),
     4    AA(65),BB1(65),CC(65),BB(65),RHSIDE(65)
          DATA XMU0/12.566371D-7/,PISQ/9.869604401/
      IF(DMIN1(R-RINNER-(ILIM1-1)*HR,RINNER+(ILIM2-1)*HR-R,
     1    Z+HEIGHT-(JLIM1-1)*HZ,-HEIGHT+(JLIM2-1)*HZ-Z))1,4,4
    4     DPSI=PSI-PSIP
      IF(TORCUR)11,11,12
   12     DPSI=-DPSI
   11 IF(DPSI-1.0D-6*DABS(PSIP))2,2,1
    1     RHS=0.0
      GO TO 3
    2     RHS=-A*(B+C*DPSI+RZEROS*XMU0*(D+E*DPSI))
    3 RETURN
      END
```

Users wishing to study different functional forms of j_ϕ need alter only the statement with label 2.

```
       FUNCTION P(PSI,R,Z)
          IMPLICIT REAL*8 (A-H,O-Z)
          COMMON/GEOM/NZP,NRP,ISYM,NHZ,NHR,INDEX,NGRID,NRED,NEQU
,NGM1,
     1    RMINOR,RMAJOR,HZ,HR,HEIGHT,RINNER,ROUTER,DELTA,DELTA1
          COMMON/PARAM/IPSIP1,IPSIP2,JPSIP1,JPSIP2,ILIM1,ILIM2,J
LIM1,
     1    JLIM2,A,B,C,PSIP,TORCUR,CALCUR,D,E,PIMU0,BZERO
      IF(DMIN1(R-RINNER-(ILIM1-1)*HR,RINNER+(ILIM2-1)*HR-R,
     1    Z+HEIGHT-(JLIM1-1)*HZ,-HEIGHT+(JLIM2-1)*HZ-Z))1,4,4
    4     DPSI=PSI-PSIP
      IF(TORCUR)11,11,12
   11 IF(DPSI-1.0D-6*DABS(PSIP))2,2,1
    1     P=0.0
      GO TO 3
    2     P=A*(D*DPSI+0.5*E*DPSI*DPSI)
      GO TO 3
   12     DPSI=-DPSI
      IF(DPSI-1.0D-6*DABS(PSIP))22,22,21
   21     P=0.0
      GO TO 3
   22     P=A*(-D*DPSI-0.5*E*DPSI*DPSI)
    3 RETURN
      END
```

The statements with labels 2 and 22 need to be altered if a different form of $p(\psi)$ is used.

```
      FUNCTION BPHI(PSI,R,Z)
         IMPLICIT REAL*8 (A-H,O-Z)
         COMMON/GEOM/NZP,NRP,ISYM,NHZ,NHR,INDEX,NGRID,NRED,NEQU
,NGM1,
     1   RMINOR,RMAJOR,HZ,HR,HEIGHT,RINNER,ROUTER,DELTA,DELTA1
         COMMON/PARAM/IPSIP1,IPSIP2,JPSIP1,JPSIP2,ILIM1,ILIM2,J
LIM1,
     1   JLIM2,A,B,C,PSIP,TORCUR,CALCUR,D,E,PIMUO,BZERO
      IF(DMIN1(R-RINNER-(ILIM1-1)*HR,RINNER+(ILIM2-1)*HR-R,
     1   Z+HEIGHT-(JLIM1-1)*HZ,-HEIGHT+(JLIM2-1)*HZ-Z))1,4,4
    4    DPSI=PSI-PSIP
      IF(TORCUR)11,11,12
   11 IF(DPSI-1.0D-6*DABS(PSIP))2,2,1
    1    BPHI=BZERO*RMAJOR/R
      GO TO 3
    2    BPHI=DSQRT(2.0*A*B*DPSI+A*C*DPSI*DPSI+RMAJOR*RMAJOR
     1   *BZERO*BZERO)/R
      GO TO 3
   12    DPSI=-DPSI
      IF(DPSI-1.0D-6*DABS(PSIP))22,22,21
   21    BPHI=BZERO*RMAJOR/R
      GO TO 3
   22    BPHI=DSQRT(-2.0*A*B*DPSI-A*C*DPSI*DPSI+RMAJOR*RMAJOR
     1   *BZERO*BZERO)/R
    3 RETURN
      END
```

The statements 2 and 22 must be altered for a different form of B_ϕ.

In RHS, both positive and negative I_p are treated by defining $\psi > \psi_p$ or $\psi < \psi_p$ within the plasma. The functions $p(\psi)$ and $F(\psi)(\equiv RB_\phi)$ are then obtained by integrating the appropriate parts of $\mu_o Rj_\phi$. The constant of integration in $p(\psi)$ is defined so that $p(\psi_p) = 0$ and the constant in $F(\psi)$ is defined so that B_ϕ at R=RMAJOR with $\psi = \psi_p$ is the vacuum B_ϕ at R=RMAJOR. The above function START is the contents of TOPEOL.START1(F) and defines an elliptic cross section plasma. The file TOPEOL.START(F)

```
      DOUBLE PRECISION FUNCTION START(X,Y)
         IMPLICIT REAL*8 (A-H,O-Z)
         COMMON/SETUP/IRSTRT,AVERT,BVERT,ALPHA,BVAL,XP,YP,RP,HP
         START=HP*(RP-DSQRT((X-XP)**2+(Y-YP)**2))
      RETURN
      END
```

defines a circular cross section plasma.

In user provided functions all the above parameters B,C,D,E,XP,YP,RP and HP are available for optional use. However, in RHS, the parameter A must be the scale factor. Apart from this, the parameters may be used quite freely.

In the straight geometry code, the composition and execution files are

```
// GROUP TEMP
// SCHEDULE S/TRIALS,8,22
// CONFG RSP=2E12,STORE=170
// EXEC TROUT
// TRIALS
// COMPOSE TEMP.CYLIND,REMJOB/CYLIND.RUN3
**OPTION TREE,ROOT
**INCLUDE CLTTHE:CYLIND.MAIN
**SEGMENT 01,A
**INCLUDE CLTTHE:CYLIND.RHS,SCRAPE,CONSTR,DATAIN,ANAL,START1
**INCLUDE CLTTHE:CYLIND.CURENT,DIRECT,TRISOL,INITAL,SPSIB,CNVER
G
**INCLUDE CLTTHE:CYLIND.TRINGL
**SUBST LEMCAL:MAGLIB.SUBS
**SEGMENT 02,A
**INCLUDE CLTTHE:CYLIND.CONTOR
**SUBST CLTTHE:CYLIND.GHOST
// ENDTRIALS
```

and

```
// GROUP TEMP
// SCHEDULE TEMP.CYLIND,50,3
// CONFG RSP=2G09,STORE=510
// FILE GRIDFL,RA,TEMP.STRAUS(N0010)
// FILE DSET97,RA,CYLIND.DSET97(S0030)
// FILE DSET99,RA,TEMP.DSET99(S0220)
// EXEC TEMP.DSET99(S0220)
```

No P and B_ϕ function routines are provided but the standard RHS is

```
      FUNCTION RHS(PSI,X,Y)
         IMPLICIT REAL*8 (A-H,O-Z)
         COMMON/GEOM/RMINOR,RMAJOR,NZP,NRP,HZ,HR,ISYM,NHZ,NHR
     1   ,HEIGHT,RINNER,ROUTER,DELTA,DELTA1
     2   ,INDEX,NGRID,NRED,NEQU,NGM1
         COMMON/PARAM/A,B,C,PSIP,IPSIP1,IPSIP2,JPSIP1,JPSIP2,
     1   TORCUR,CALCUR,D,E,PIMU0,ILIM1,ILIM2,JLIM1,JLIM2
         COMMON/SOLVE/NSTEPS,NOIT,OMEGA,EPS,PHI(65,65),PHINEW(6
 5,65),
     1   ITNO,ICONVS,NUPDAT,ICONVP,PHIVI1(65),PHIVI2(65),
     2   PHIVJ1(65),PHIVJ2(65),IPOINT,ICURNT
     3   ,CRZ,CRZ0,OMEGA1,SOURCE(65,65)
     4   ,PHITWO(65,65)
     5   ,AA(65),BB1(65),CC(65),BB(65)
     6   ,RHSIDE(65)
         DATA XMU0/12.566371D-7/,PISQ/9.869604401/
      IF(DMIN1(X+HEIGHT-(ILIM1-1)*HZ,-HEIGHT+(ILIM2-1)*HZ-X,
     1   Y-RINNER-(JLIM1-1)*HR,RINNER+(JLIM2-1)*HR-Y))1,4,4
    4    DPSI=PSI-PSIP
```

```
      IF(TORCUR)11,11,12
   12    DPSI=-DPSI
   11 IF(DPSI-1.0D-6*DABS(PSIP))2,2,1
    1    RHS=0.0
      GO TO 3
    2    RHS=-A*(B+C*DPSI)
    3 RETURN
      END
```

Statement 2 should be altered for a different functional form.

The 2 START files are

```
      DOUBLE PRECISION FUNCTION START(X,Y)
         IMPLICIT REAL*8 (A-H,O-Z)
         COMMON/SETUP/AVERT,BVERT,ALPHA,BVAL,XP,YP,RP,HP
         START=RP*(1.0-X*X/XP/XP-Y*Y/YP/YP)+RP
      RETURN
      END
```

and

```
      DOUBLE PRECISION FUNCTION START(X,Y)
         IMPLICIT REAL*8 (A-H,O-Z)
         COMMON/SETUP/AVERT,BVERT,ALPHA,BVAL,XP,YP,RP,HP
         START=HP*(RP-DSQRT((X-XP)*(X-XP)+(Y-YP)*(Y-YP)))
      RETURN
      END
```

Again, the parameters B,C,D,E,XP,YP,HP and RP are available for optional use, although A must be the scale factor of the current density.

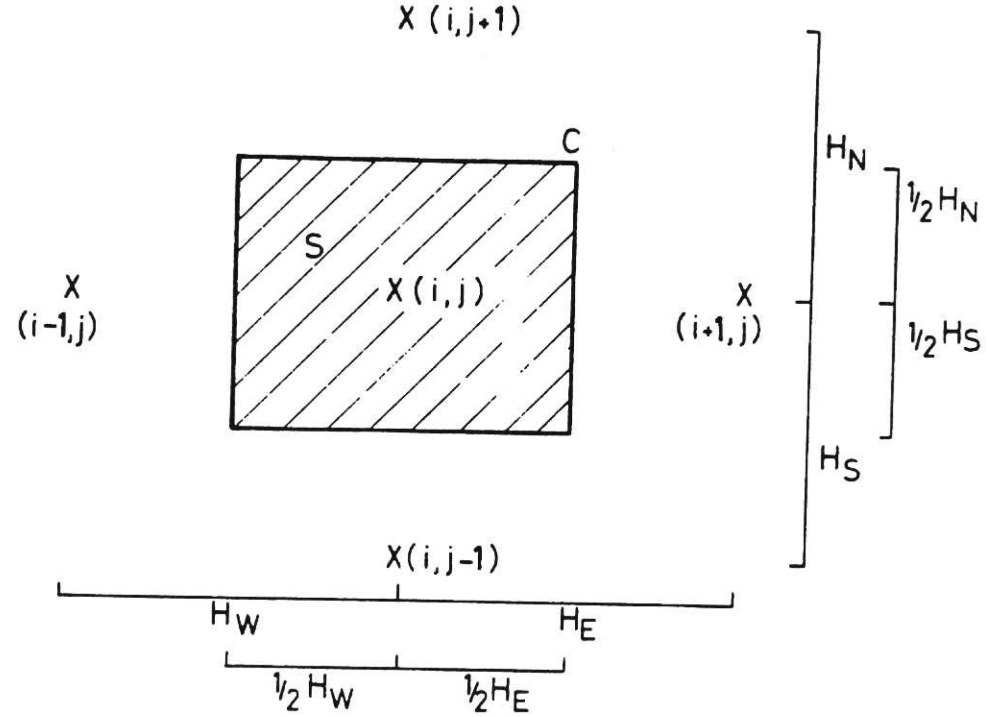

Fig.1 Mesh cell used for constructing the difference equations.

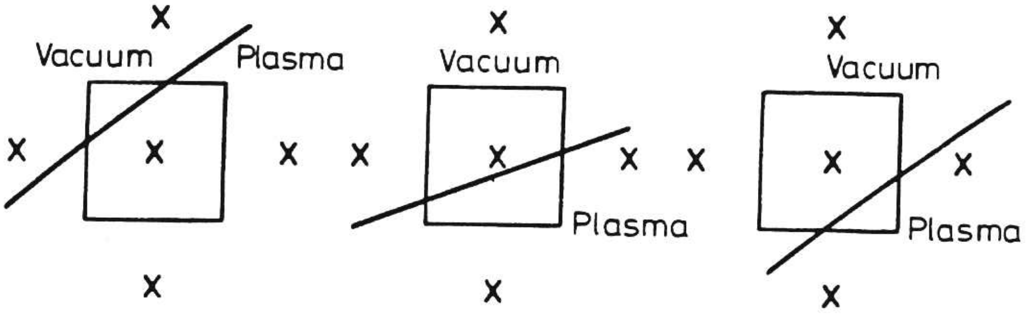

Fig.2 Intersections of the interface with a mesh cell.

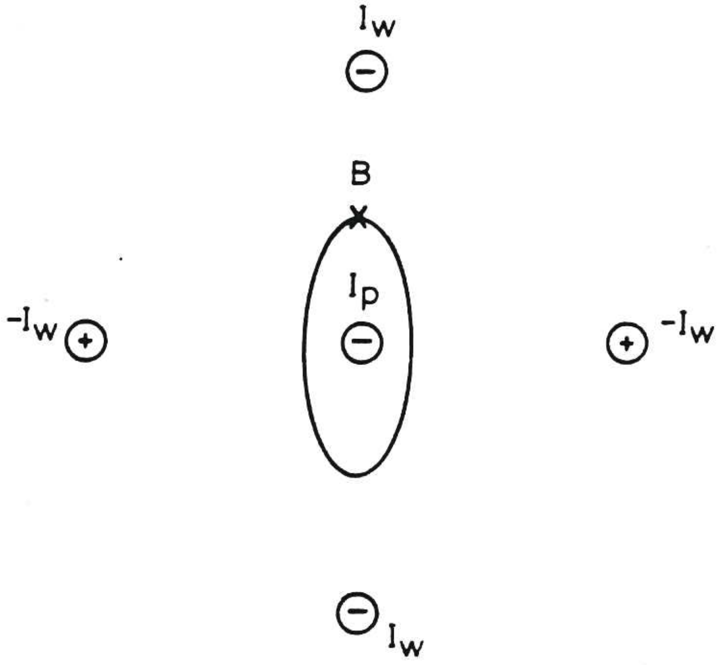

Fig.3 Elliptic plasma in a quadrupole field.

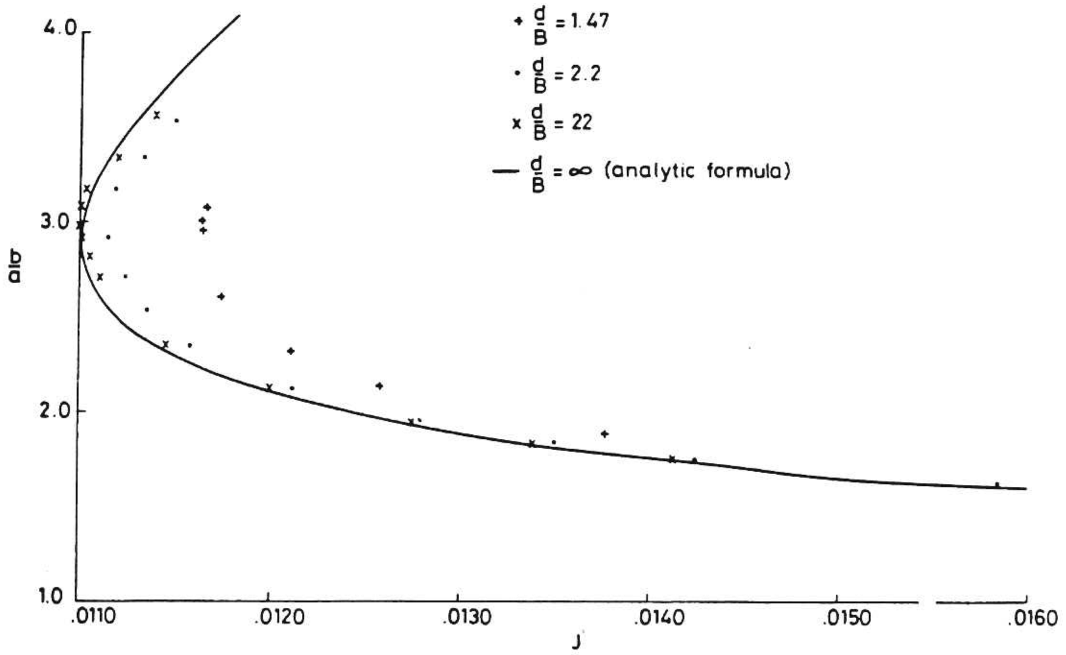

Fig.4 Plots of current density, j, against plasma b/a, showing results for various ratios of plasma height to wire separation together with the asymptotic results.

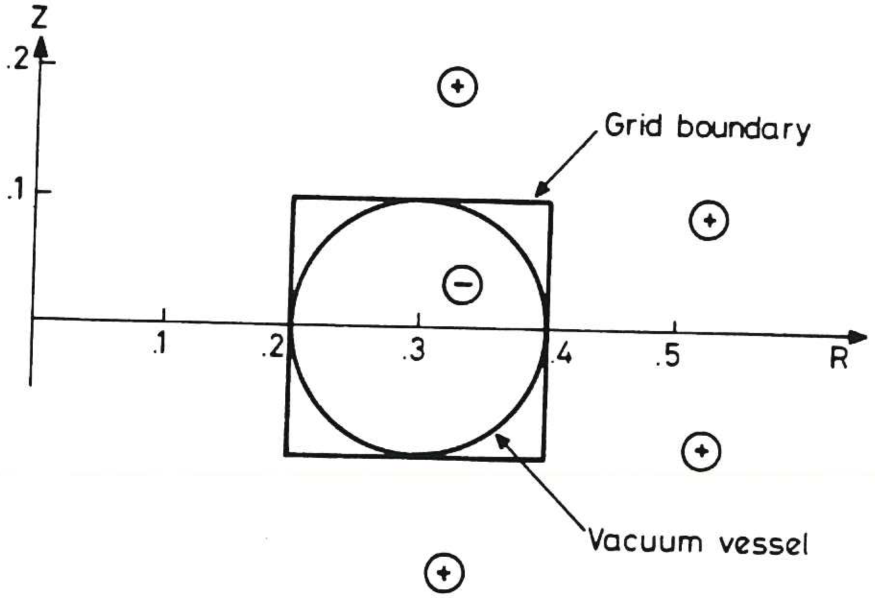

Fig.5 TOSCA configuration used in the computations.

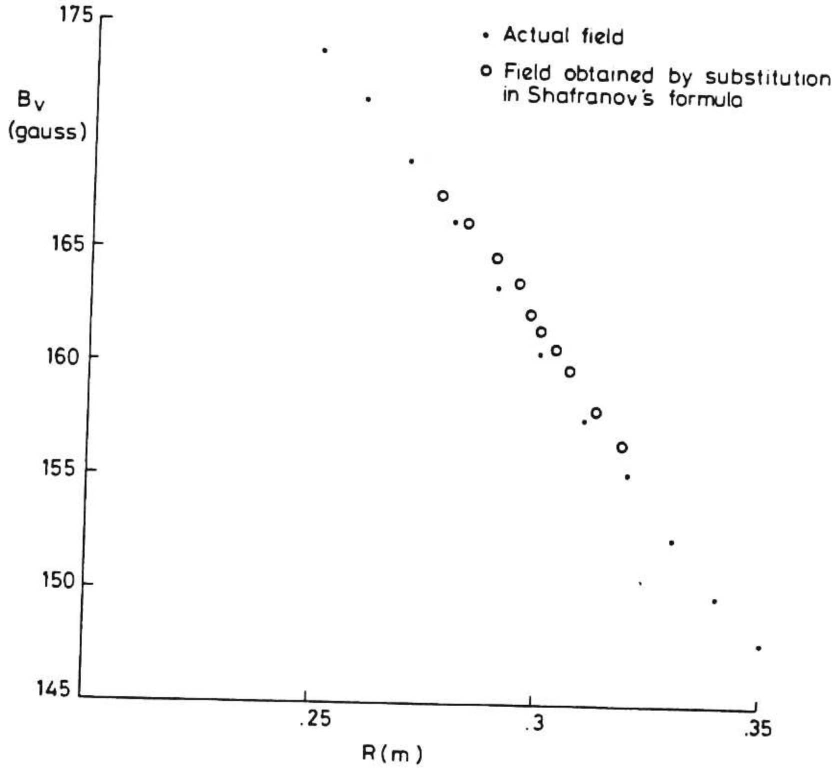

Fig.6 Comparison between the TOSCA vertical field and the Shafranov vertical field obtained from the computations.

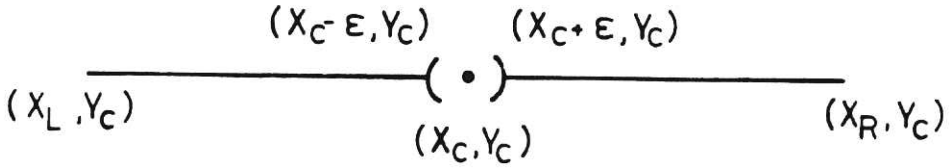

Fig.7 Notation for subtracting the singularity in the boundary integral.

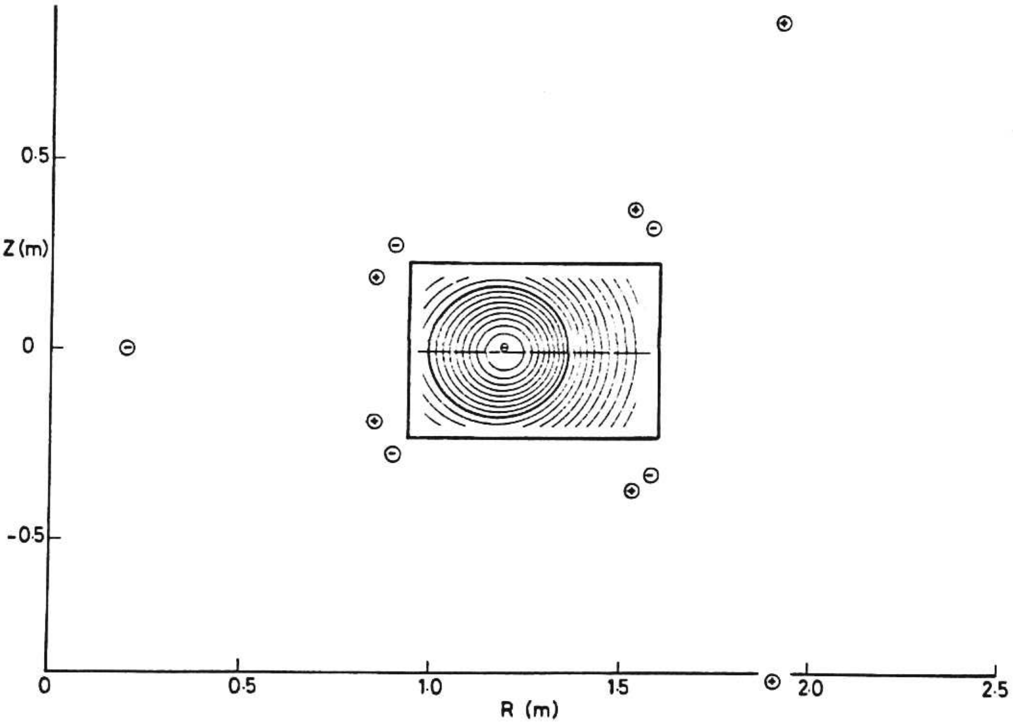

Fig.8 A typical DITE equilibrium showing the winding data used in the computations. The dashed line is the plasma boundary.