I-10927

LBL-10070
UC-20g

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Accelerator & Fusion Research Division

BL--10070

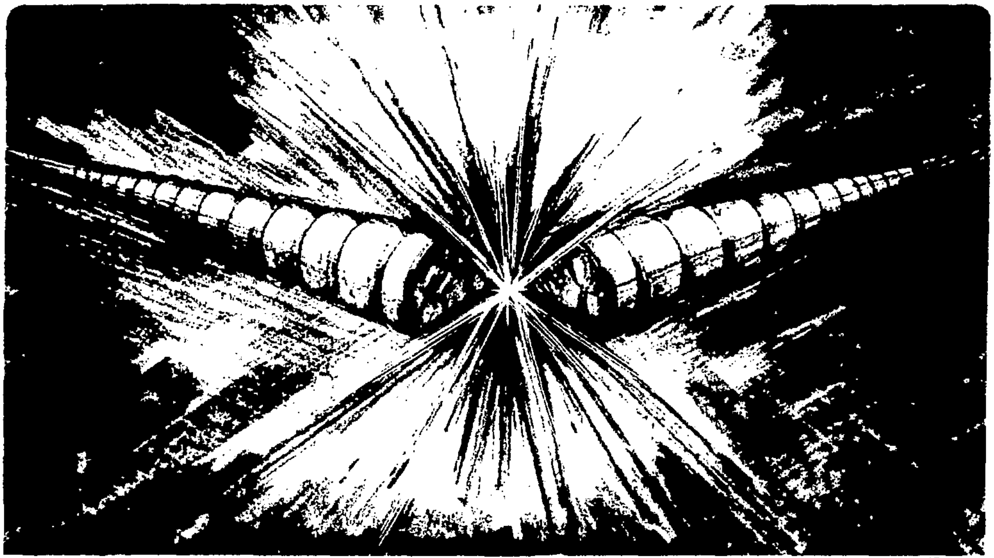
L383 016916

A 2-D SKIN-CURRENT TOROIDAL-MHD-EQUILIBRIUM CODE

B. Feinberg, R.A. Niland, J. Coonrod,
and M.A. Levine

September 1982

NOTICE
PORTIONS OF THIS REPORT ARE ILLEGIBLE.
It has been reproduced from the best available copy to permit the broadest possible availability.



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

MASTER

A 2-D Skin Current Toroidal-MHD-Equilibrium Code[*]

B. Feinberg, R. A. Niland[†], J. Coonrod[‡], and M. A. Levine

Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Septebmer, 1982

Abstract

A two-dimensional, toroidal, ideal MHD skin-current equilibrium computer code is described. The code is suitable for interactive implementation on a minicomputer. Some examples of the use of the code for design and interpretation of toroidal cusp experiments are presented.

---

[*] This work was supported by the Director, Office of Energy Research, Office of Fusion Energy, Applied Plasma Physics Division, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098.

[†] Present address: University of Sydney, Physics Dept., Sydney, Australia
[‡] Present address: Princeton University, PPL, Princeton, NJ 08540

I. Introduction

An MHD equilibrium code is extremely useful for the design and interpretation of toroidal plasma containment experiments. Even an ideal skin current model can help determine the location of the field coils for a non-circular plasma, as well as the plasma boundary. The two dimensional time dependent MHD code PATENT[1] was initially used to study toroidal magnetic cusp formation and equilibrium.[2] Difficulties in modeling the boundary during the formation phase led to the development of this skin current equilibrium code. The ease of running a skin current model on a minicomputer in the laboratory allows the experimenter to make rapid changes in the configuration and enhances understanding of the data.

An interactive computer code has been written to model a non-circular, toroidal, β=1 plasma equilibrium. The code uses a skin current model, with discrete coils representing the skin currents, and is suitable for use on a mini-computer (e.g.: PDP 11-34). Design of two Tormac (Toroidal Magnetic Cusp) experiments, TVB[3] and Tormac P-1[4], and data interpretation of Tormac P-1[5] were aided by the use of this code. The code is simple to use, and with less than an hour's work the plasma equilibrium position can be obtained, starting with the coil configuration and currents, and the plasma pressure. When designing an experiment this procedure can be iterated to provide the desired plasma shape.

The problem can be described as follows. A set of external coils, both toroidal and poloidal, surrounds the plasma volume. Coil locations and currents are given and may yield an equilibrium plasma configuration. Plasma currents are inductively coupled to the external coils, while the plasma position is determined by the balance of plasma and magnetic pressures. The code determines the equilibrium plasma position, if it exists, depending on the initial current in the plasma and the desired plasma pressure.

The plasma is modeled by a set of toroidal current carrying conductors approximately evenly spaced along the plasma border. These currents are determined by the poloidal flux function, ψ , on the plasma boundary and by the external coils. Once the currents have been found, the boundary points are moved semiautomatically to satisfy pressure balance. The process is iterated until a satisfactory solution is obtained.

The solution procedure is described in Section II. Section III gives some details of program verification, while two examples of the use of the code are provided in Section IV. Appendix I gives the source files in Fortran IV, while Appendix II includes a user's guide.

II. Solution Procedure

The skin current model is based on the static, ideal magnetohydrodynamic (MHD) equations[6] for force balance, along with the relevant steady state Maxwell's equations.

$$\bar{\nabla} P = \bar{J} \times \bar{B}/c \qquad (1)$$

$$(4\pi/c)\, \bar{J} = \bar{\nabla} \times B \qquad (2)$$

$$\bar{\nabla} \cdot \bar{B} = 0 \qquad (3)$$

Equations (1) and (2) yield

$$\bar{\nabla} P = - \bar{\nabla} \left(\frac{B^2}{8\pi}\right) + \frac{1}{4\pi} (\bar{B} \cdot \bar{\nabla})\bar{B} \qquad (4)$$

Since the radius of curvature of the field lines, in our case, is much greater than the thickness of the current layer over which the magnetic field changes, equation (4) becomes the usual pressure balance condition,

$$P + \frac{B^2}{8\pi} = \text{constant} \qquad (5)$$

Equation (3) implies that we can define the poloidal flux function $\psi = RA_\phi$, where A_ϕ is the toroidal component of the magnetic vector potential.

The iterative procedure used to solve the problem is illustrated in Fig. 1. It basically consists of two parts: computing the plasma currents, which then enables the calculation of the magnetic pressure, and moving the plasma boundary. Each time the boundary is moved, a new calculation of the currents is performed. The magnetic pressures at the boundary are presented after a user supplied number of repetitions, at which time the user can continue the iteration or display the configuration.

Step A -- The user inputs the display grid size and location. This enables the display of the entire plasma or any desired region. External coil locations and currents, and the position of any fixed closed conductors are also input. Note that fixed conductors are represented by discrete toroidal loops.

Step B -- In a like manner the positions and number of toroidal coils representing the plasma are input, as well as the ψ value on the plasma boundary. More detailed instructions on setting the ψ value are included in Step C. If the vacuum field alone is desired, one can branch directly to Step C.

Step C -- The toroidal currents induced in the plasma and any other closed loops are calculated in this step. The poloidal flux is given by

$$\Phi = \int B \cdot d\bar{s} = 2\pi \psi \qquad (6)$$

The mutual inductance, $M_{ij} = d\Phi_i/dI_j = 2\pi\psi_i/I_j$ can be calculated at each coil location i for each coil j. Elliptic integrals(7) are used to calculate ${}_i$ at location i for a unit current in coil j. This gives the mutual inductance between the two coils at i and j. A second order polynomial approximation is used for the elliptic integrals themselves(8). Once all the mutual inductances are known, one can, therefore, write the matrix equation

$$\sum_j M_{ij} \, I_j = 2\pi\psi_i \qquad (7)$$

The summation is over all currents I_j and inductances M_{ij}, and yields the poloidal flux function ψ at each position i. Since the value of ψ at the plasma surface is input, ψ at the coil locations representing the plasma boundary is known. The poloidal flux function, ψ, determines the total toroidal current present in the plasma. If there is no plasma current with the external coils removed, then $\psi = 0$ at the boundary.

The option of setting $\psi \neq 0$ in the problem was included so that one could easily model the final equilibrium of a preionized plasma with a toroidal current present before the main external coils were turned on. If one knows the current and approximate extent of the preionized plasma, one can put plasma coils at that boundary, with no external coils, and change ψ until one has the desired total current. Since the equations are linear, scaling a single trial ψ will work. All other toroidal conductors in the problem (e.g.: a cylindrical center conductor along the Z axis represented by a row of circular loops) are assumed to start off with no current, meaning $\psi = 0$. The problem is therefore reduced to inverting the mutual inductance matrix, which is accomplished using a Choleski decomposition method.[9]

It should be noted that the inductance matrix is ill conditioned. This means that small changes in ψ_i will produce large changes in I_j. Another way to express this is that there is a large collection of currents I_j which come close to satisfying the equation. Physically, this can be seen by looking at two coils close together. The exact division of currents between them is unimportant, only the sum of the two currents matters in determining ψ. Fortunately we are only interested in using the entire collection of currents to determine the field magnitude, and not in the values of the individual currents.

Step D -- The magnetic pressure at the plasma surface is calculated in this step. Radial and axial components of the magnetic field at each plasma boundary coil location are summed over all the current carrying conductors. To avoid the large perturbation in the field due to the current in the coil at which the field is being measured, that current is temporarily set to zero. This procedure eliminates some of the error introduced by having discrete coils instead of a distributed surface current. A numerical test was made of the effects of setting one coil's current equal to zero. The change in field at any one coil, measured by using graphs of $|B|$ vs R and $|B|$ vs Z on a 10 cm x 10 cm grid, was less than 2-1/2 percent. All the plasma coils from Fig. 3b were tested in this manner. Considering the simplicity of the discrete coil model this accuracy was deemed sufficient. The ill conditioned state of the inductance matrix causes large variations in coil currents for small perturbations in the plasma boundary position. At each coil location, however, the poloidal field is due to the currents in all the coils, and this sum is not greatly disturbed by the ill conditioned matrix. As a check, the plasma currents of Fig. 3b were changed by altering ψ at the boundary. The total plasma current was changed by 5 percent, resulting in a maximum individual coil current change of 24 percent. The $|B|$ value at any coil was changed, however, by less than 0.5 percent.

The toroidal field is calculated at each plasma surface location, using the value of axial current supplied initially. Diamagnetism of the plasma is assumed to result in the poloidal currents needed to prevent the toroidal field from entering the plasma. A simple argument shows, at least for the Tormac case, that the toroidal beta has little influence on the plasma equilibrium shape. For the more realistic case with toroidal field present in the plasma, the pressure balance condition can be written as

$$P_o = nkT = (B_T^2 + B_p^2 - B_{Ti}^2)/8\pi \qquad (8)$$

where B_{Ti} is the internal toroidal field. Since all currents and pressure drops in this model are at the surface, both the internal toroidal field B_{Ti} and the external toroidal field B_T are inversely proportional to R, the major radius. At the innermost radial boundary position, R_o, the poloidal field, B_p, is negligible in a bicusp such as Tormac. One can therefore write

$$8\pi\, P_o = (B_{T_o}^2 - B_{Ti_o}^2)$$

Equation 8 can be written as

$$8\pi\, P_o = (B_{T_o}^2 - B_{Ti_o}^2) \quad \frac{R_o^2}{R^2} + B_p^2$$

Therefore

$$B_p^2 = 8\pi\, P_o(1 - \frac{R_o^2}{R^2})$$

and B_p^2 is independent of the internal toroidal field B_{Ti}. For simplicity the internal toroidal field is therefore assumed to be zero in this code. The magnetic pressure at each location is the sum of the squares of the poloidal and toroidal field components at that location, divided by 8π.

Step E -- Once the magnetic pressures are calculated, they are compared at each boundary location with the desired plasma pressure, which is input at this point. The boundary currents are then displaced normal to the surface by a distance proportional to a user supplied step size times the difference in pressures. For each coil position the outward normal is calculated by obtaining the line connecting two adjacent boundary points, one on each side of the point of interest, and using the perpendicular to this line which intersects the original coil position. Each outward perpendicular is normalized to a unit length. Boundary coils are moved outward if the magnetic pressure is less than the desired plasma pressure, and inward if the magnetic pressure is too high. The magnitude of the displacement is proportional to the pressure difference and the user supplied step size. Next, symmetry about the midplane is assumed. So all boundary coil positions are averaged with their mirror image coil positions to minimize the accumulation of errors. Steps C, D, and E are iterated the number of times desired, and the plasma boundary coil positions and magnetic pressures are then listed. The user may go on to Step F displaying the plasma, input a change in a coil location directly, or continue the convergence process with a new step size and repetition number.

In addition to this semiautomatic method of boundary coil movement, provisions have been made for direct input of new coil positions. This is especially useful for keeping the coils approximately evenly spaced along the boundary. Since the coils are meant to represent a surface current, even spacing between coils is desired. The distance from one coil to the next is displayed along with the pressures to help the user adjust the spacing.

Step F -- A variety of options are available for displaying the plasma and field configuration. Two basic types of displays are available, contour plots and graphs. Contours of constant ψ, or poloidal magnetic field lines, are available as well as contours of constant $|B|$. The user determines the range of ψ or $|B|$ to be displayed, as well as the number of contours in that range.

Graphs of $|B|$ as a function of the radius R and a particular axial position Z can be plotted, along with $|B|$ as a function of Z at a particular radius R. All of these displays can be presented for the vacuum field as well. When the plasma is displayed, the boundary coil locations and the vacuum coils are shown by "X"'s. It should be noted that any region of space can be displayed. The program calculates the ψ and $|B|$ values on a 40x40 grid. This R, Z grid has an arbitrary origin and grid spacing which enables the user to concentrate on the entire configuration, or any desired portion thereof. Note that ψ is calculated as described in Step C, through the use of a second order polynomial approximation for the elliptic integrals.[7,8] $|B|$ is calculated from the ψ values using $\overline{B} = \overline{\nabla} \times \overline{A}$, with the space between grid points as the step size. A central difference formula is used, with the error fourth order of the step size.[10]

III. Validation

An important part of designing a computer code is testing the validity of the results. Two methods were used to verify this code's solutions. Simple external coil configurations were input, and the fields produced were checked against analytic solutions at a number of grid points. Since the same method of calculating the fields is used repeatedly in the program, this check is essential.

As the configuration can be examined during convergence, the magnetic pressures at each boundary point can be printed. If the code is working properly these pressures should converge to the desired plasma pressure, which is the case. A check is made by starting with two different sets of initial conditions and observing whether they converge automatically to the same final configuration. Figures 2a and 2b show two quite different initial configurations. The final solutions are shown in Figures 3a and 3b, respectively. The convergence to the same solution is quite good for these cases, taking into account the uneven coil spacing. With manual spacing of the coils parallel to the boundary the solutions would converge even more closely. It should be recalled that this program is intended as an experimental design and data interpretation aid, and thus agreement to a few percent in pressure is sufficient. Note that the agreement is in the boundary location and boundary pressures, and not the individual coil currents. This is due to the ill conditioned inductance matrix. Since the coils are actually fictitious, this discrepancy is unimportant.

IV. Applications

This computer program has been used extensively in our group, principally by experimenters wishing to design or modify experiments, or to assist in data analysis and interpretation. Two new experiments were designed, Tormac VB[3] and Tormac P1[4]. Tormac P1 was built and operated, and the computer code played an integral role in the interpretation of the experimental observations.[5]

Tormac VB design presented some special problems. The goal was to build an experiment about the size of Tormac IV[11] (15 cm major radius) but with the vessel walls removed from the cusp locations. Fig. 4 shows the plasma, poloidal field lines, and vessel configuration. Since the plasma modifies the field, and the vessel extensions must be small in width to allow the proper coil placements, calculations were made to determine the desired vessel shape and coil currents. Fig. 5 shows the vacuum field for comparison.

A second example of the use of this code was in data interpretation.[5] Particle flux out the cusps was measured in Tormac P-1. As shown in Fig. 6, tracing the field lines back to the measured plasma location gives information about the width of the plasma boundary (or sheath), and the total plasma current. For comparison, Fig. 7 shows the vacuum field configuration. The extensions of the cusp field lines from the location of the main body of the plasma do not fall near the measured flux for the vacuum fields. The code was also used to aid in the interpretation of interferometer data. Modeling the plasma by adjusting the boundary and ψ to conform with the observed radial position enabled an estimate of the plasma width in the axial direction. This was used to determine the number density from the line density.

V. Conclusion

An interactive skin current equilibrium code has been described. This code models idealized MHD equilibrium configurations of non-circular toroidal plasmas with sharp boundaries. Experimental design and data interpretation have been aided through the use of this code, which is quick and simple to implement on a minicomputer. Experimental design changes can therefore be rapidly planned by the experimental group.

References

1. S. C. Jardin, J. L. Johnson, J. M. Greene, R. C. Grimm, J. Comp. Phys. 29, 101 (1978).
2. A. Sleeper, H. L. Berk, S. C. Jardin, Bull. Am. Phys. Soc. 23, 859 (1978).
3. B. Feinberg, J. Coonrod, M. A. Levine, Bull. Am. Phys. Soc. 23, 859 (1978).
4. L. Soroka, M. A. Levine, B. R. Myers, Bull Am. Phys. Soc. 23, 860 (1978).
5. P. A. Pincosy, B. R. Myers, M. A. Levine, B. Feinberg, R. A. Niland, L. Soroka, Lawrence Berkeley Laboratory Report LBL-10453, also submitted to Phys. Fluids.
6. F. F. Chen, Introduction to Plasma Physics, (Plenum Press, New York, 1974), p. 175.
7. W. R. Smythe, Static and Dynamic Electricity, (McGraw Hill, New York, 1950), p.270.
8. J. Hart, Computer Approximations, (John Wiley, New York, 1968), p. 154.
9. D. Kershaw, J. Comp. Phys., 26, 43 (1978).
10. M. G. Salvadori, M. L. Baron, Numerical Methods in Engineering (Prentice Hall, Englewood Cliffs, N. J., 1961), p.86.
11. M. Greenwald, Ph. D. thesis, Univ. of Cal., Berkeley (1978); J. W. Coonrod, Ph. D. Thesis, Univ. of Cal., Berkeley (1978).

APPENDIX I

```
C       ***** EPLAPC.FTN *****
C
C--------MAIN PROGRAM TO CALCULATE AND DISPLAY PSI AND MOD B
C--------FOR AN ARBITRARY SET OF CURRENT LOOPS. USE PLANTRE.ODL TO LINK
C--------THIS PROGRAM  WITH SUBPROGRAMS EPSICAL,BBFF,ECURR,
C--------BDRAW AND EPLIN, AS WELL AS GRAF AND CONTURB.
C--------THIS PROGRAM HAS THE CAPABILITY OF MODELING THE PLASMA
C--------BOUNDARY BY COILS WHOSE CURRENTS ARE CALCULATED AS A FUNCTION
C--------OF POSITION IN THE SUBPROGRAM EPLIN.
C--------THE PROGRAM USES A 40 X 40 GRID, WITH AN ARBITRARY STEP
C--------SIZE IN EACH DIRECTION, AND ARBITRARY STARTING POSITIONS.
        PROGRAM EPLAPC
        REAL IL(50)
        COMMON/BLK1/RL(50),ZL(50),IL,NL,IPLAME(8),PSI0
        COMMON/BLK2/CR(50),CZ(50),CI(50),NC,INAME(8)
        COMMON/BLK3/ACR(50),ACZ(50),ACI(50),NAC,ICCAME(8)
        COMMON/BLK5/IPSAME(8),CURZ,IFLG,JFLG
        COMMON/BLK6/RSTEP,ZSTEP,IRSTRT,IZSTRT
        COMMON/BLK7/P(50),RN(50),ZN(50),DLB(50),P1
        DATA IEX,ICO,IPL,ICP,NF,NI/2HEX,2HCO,2HPL,2HCP,2HNF,2HNI/
        DATA NO,IDI,IBR/2HNO,2HDI,2HBR/
        IGS=81
        IBL=7
        IPLFLG=0
C
C--------INPUT GRID INFORMATION
30      CALL TVCL
        TYPE 35,IGS
35      FORMAT(1X,A1,'INPUT START R, END R, START Z, END Z (CM,I3:0,40,
     1 -20,20)')
        READ(5,40,ERR=30)IRSTRT,IREND,IZSTRT,IZEND
40      FORMAT(4I4)
        RSTEP=(IREND-IRSTRT)/40.
        ZSTEP=(IZEND-IZSTRT)/40.
C--------INPUT CENTER CONDUCTOR
50      TYPE 52
52      FORMAT(' INPUT CENTER CONDUCTOR'/)
        IFLG=0
        CALL COIL(1,ICCAME,ACR,ACZ,ACI,NAC)
        IF(NAC.LE.0)IFLG=1
C
C--------INPUT EXTERNAL COILS
        TYPE 62
62      FORMAT (' INPUT EXTERNAL COILS'/)
        CALL COIL(2,INAME,CR,CZ,CI,NC)
C
C--------INPUT Z CURRENT
70      TYPE 75
75      FORMAT(' ENTER Z CURRENT (ABAMPS) (0:1.E4) ',$)
        READ(5,76,ERR=70)CURZ
76      FORMAT(E11.3)
```

```
C
C--------INPUT THE PSI FILE
         JFLG=0
30       TYPE 62
62       FORMAT(' INPUT PSI FILENAME OR "NO"?',$)
         READ(5,64,ERR=30)IPSAME
64       FORMAT(8A2)
         IF(IPSAME(1).NE.NO)JFLG=1
C
C--------BRANCHING SECTION
65       TYPE 67,IGS
67       FORMAT(1X,1A1,'PL PLASMA,CP CHANGE PLASMA,NE=NEW
         1 FILE',' NI NEW I CURRENT,CO=COILS ONLY,EX=EXIT ',$)


         READ(5,22,ERR 65)NCMD
22       FORMAT(1A2)
         CALL IVCL
         IF(NCMD.NE.IEX)GOTO 35
         STOP '---EXIT'
35       IF(NCMD.EQ.IPL)GOTO 150
         IF(NCMD.EQ.ICP)GOTO 200
         IF(NCMD.EQ.INE)GOTO 30
         IF(NCMD.EQ.INI)GOTO 70
         IF(NCMD.EQ.ICO)GOTO 250
C
C--------INPUT PLASMA & GET CURRENTS
150      CALL PLIN1
         IPLFLG=1
155      TYPE 160,IGS
160      FORMAT(1X,1A1,'DI DISPLAY CONFIGURATION,BR=BRANCH ',$)
         READ(5,22,ERR 155)NCMD
         IF(NCMD.EQ.IBR)GOTO 65
         CALL DRAW(1)
         GOTO 155

C--------CHANGE PLASMA & GET CURRENTS & DRAW
200      IF(IPLFLG.EQ.0)TYPE 210,IGS
210      FORMAT(1X,1A1,'ENTER PLASMA BEFORE CHANGING PLASMA',$)
         IF(IPLFLG.EQ.0)GOTO 65
         CALL PLIN1
         GOTO 155
C
C--------COILS ONLY
250      CALL PLIN2
255      TYPE 160,IGS
         READ(5,22,ERR 255)NCMD
         IF(NCMD.EQ.IBR)GOTO 65
         CALL DRAW(1)
           GOTO 255
         END
```

```
C
C
C-------SUBROUTINE TO ENTER COILS
C
        SUBROUTINE COIL(M,NAME,RL,ZL,IL,NL)
C--     M=0 > INPUT R,Z,I FILE
C--     M=1 > INPUT R,Z FILE
C--     M=2 > OUTPUT R,Z FILE
C--     M=3 > OUTPUT R,Z,I
C
        REAL IL(50),RL(50),ZL(50)
        INTEGER NAME(8),INAME(8)
        DATA NO/2HNO/
        IGS=31
        IF(M.EQ.2)GOTO 300
        IF(M.EQ.3)GOTO 200
C
C-------INPUT DATA FILE
10      TYPE 15
15      FORMAT(' COIL DATA FILE?(NAME OR NO) ',$)
        READ(5,20,ERR=10)INAME
20      FORMAT(8A2)
C
C-------BRANCH IF NO FILE
        IF(INAME(1).EQ.NO) GO TO 50
          DO 22 I=1,8
22        NAME(I)=INAME(I)
        CALL ASSIGN(1,NAME,16)



C
C-------BRANCH IF M=1 (JUST R & Z)
        IF(M.EQ.1)GOTO 40
        READ(1,25,ERR=10)NL
25      FORMAT(I3)
        READ(1,30,ERR=10)(RL(I),ZL(I),IL(I),I=1,NL)
30      FORMAT(3F10.3)
35      CALL CLOSE(1)
        RETURN
C
C-------JUST R & Z
40      READ(1,25,ERR=10)NL
        READ(1,45,ERR=10)(RL(I),ZL(I),I=1,NL)
45      FORMAT(2F10.2)
        CALL CLOSE(1)
        RETURN
C
```

```
C--------INPUT DATA AND BRANCH IF JUST R AND Z
50      IF(M.EQ.1)GOTO 250
        TYPE 55
55      FORMAT(' ENTER NUMBER OF COILS (EG:19) ',$)
        READ(5,25,ERR=50)NL
56      TYPE 57
57      FORMAT(' ENTER R(CM),Z(CM),I(ABAMP) (EG:5.21,10.3,50.5) ',$)
        READ(5,40,ERR=56)(RL(I),ZL(I),IL(I),I=1,NL)
C
C--------OUTPUT DATA FILE
200     TYPE 210
210     FORMAT(' OUTPUT DATA FILENAME OR "NO'? ',$)
        READ(5,20,ERR=200)INAME
        IF(INAME(1).EQ.NO) RETURN
          DO 212 I=1,8
212       NAME(I)=INAME(I)
        CALL ASSIGN(1,NAME,16)
        WRITE(1,25)NL
        WRITE(1,30)(RL(I),ZL(I),IL(I),I=1,NL)
        CALL CLOSE(1)
        RETURN
C
C--------INPUT R AND Z DATA
250     TYPE 55
        READ(5,25,ERR=250)NL
        IF(NL.EQ.0)RETURN
255     TYPE 260
260     FORMAT(' ENTER R(CM),Z(CM) (EG:14.32,1.2) ',$)
        READ(5,45,ERR=255)(RL(I),ZL(I),I=1,NL)
C
C--------OUTPUT R AND Z DATA FILE
300     TYPE 210
        READ(5,20,ERR=300)INAME
        IF(INAME(1).EQ.NO)RETURN
          DO 310 I=1,8
310       NAME(I)=INAME(I)
        CALL ASSIGN(1,NAME,16)
        WRITE(1,25)NL
        WRITE(1,45)(RL(I),ZL(I),I=1,NL)
        CALL CLOSE(1)
        RETURN
        END
```

```
C
C
C--------FUNCTION SUBROUTINE FOR ELLIPTIC INTEGRAL K
C--------REF. J. HART. COMPUTER APPROX.,1968.JOHN WILEY,NY,P.154
        FUNCTION ELK(X)


        DATA A,B,C/1.3862944,.11196966,.072532303/
        DATA E,F,G/.5,.12134863,.028874723/
        Y=1.-X
        IF(Y.LT.1.E-4)Y=1.E-4
        P=A+Y*(B+Y*C)
        Q=E+Y*(F+Y*G)
        ELK=P-ALOG(Y)*Q
        RETURN
        END
C
C
C--------FUNCTION SUBROUTINE FOR ELLIPTIC INTEGRAL E
C--------REF. J. HART, COMPUTER APPROX.,1968,JOHN WILEY,NY,P.154
        FUNCTION ELE(X)
        DATA A,B,C/1.,.46371058,.10773575/
        DATA F,G/.24527797,.041253211/
        Y=1.-X
        IF(Y.LT.1.E-4)Y=1.E-4
        P=A+Y*(B+Y*C)
        Q=Y*(F+Y*G)
        ELE=P-ALOG(Y)*Q
        RETURN
        END
```

```
C          ***** PLIN.FTN *****
C
C--------SUBROUTINE TO CALCULATE CURRENTS TO MODEL PLASMA
C--------USER DETERMINES COIL LOCATIONS AND PLI DETERMINES CURRENTS
C
      SUBROUTINE PLI(M1)
C--   M1=0 > INPUT PLASMA
C--   M1=1 > CHANGE PLASMA
C--   M1=2 > VACUUM (CENTER CONDUCTOR CURRENTS)
C--   IFLG=0 > CENTER CONDUCTOR PRESENT
C--   IFLG=1 > NO CENTER CONDUCTOR
C
      REAL IL(50),R(50),RN1(50),ZN1(50),DLE(50)
      INTEGER KON(50),KNOT(50)
      COMMON/BLK1/RL(50),ZL(50),IL,NL,IPLAME(8),PSI0
      COMMON/BLK2/CR(50),CZ(50),CI(50),NC,INAME(8)
      COMMON/BLK3/ACR(50),ACZ(50),ACI(50),NAC,ICCAME(8)
      COMMON/BLK5/IPSAME(8),CURZ,IFLG,JFLG
      COMMON/BLK7/R,RN1,ZN1,DLE,P1
      COMMON/BLK8/BR,BZ
      DATA PI,IFG,NFG/3.14159,0,0/
      IRL=7
C
C--------CALCULATE CENTER CONDUCTOR CURRENTS (NO FILTER CONS'S) ONLY
      IF(M1.EQ.2 .AND. IFLG.EQ.0)CALL CURR(ACR,ACZ,ACI,NAC,NAC,0.
      IF(M1.EQ.2)RETURN
C
C--------TO CHANGE PLASMA GO TO 40
      IF(M1.EQ.1)GOTO 40
C
C--------INPUT EDGE VALUE OF PSI
5     TYPE 10
10    FORMAT('$INITIAL PSI AT PLASMA SURFACE (EG:1.25) ')
        READ(5,12,ERR=5)PSI0
12    FORMAT(F11.0)
C
C--------INPUT THE PLASMA
        DO 15 I=1,50
        RL(I)=0.
        ZL(I)=0.
15      IL(I)=0.
      CALL COIL(1,IPLAME,RL,ZL,IL,NL)
C
C--------CALCULATE PLASMA AND CENTER CONDUCTOR CURRENTS
C--   PLASMA
20    IF(IFLG.EQ.1)CALL CURR(RL,ZL,IL,NL,NL,PSI0)
      IF(IFLG.EQ.1)GOTO 42
C
```

```
C--       PLASMA & CENTER CONDUCTOR
          NT=NL+NAC
            DO 25 I=1,NAC
              K=NL+I
              RL(K)=ACR(I)
              ZL(K)=ACZ(I)
25            CONTINUE
          CALL CURR(RL,ZL,IL,NT,NL,PSI0)
            DO 30 I=1,NAC
            K=NL+I
            ACI(I)=IL(K)
            RL(K)=0.
            ZL(K)=0.
            IL(K)=0.
30          CONTINUE
C



C--------CALCULATE PLASMA PRESSURES
40          DO 50 K=1,NL
            K1=K+1
            K2=K-1
            IF(K.EQ.1)K2=NL
            IF(K.EQ.NL)K1=1
            P(K)=0.
            ZM=ZL(K)
            RM=RL(K)
            TEMPI=IL(K)
            IL(K)=0.
            CALL BPSIC0(ZM,RM,RL,ZL,IL.NL,K)
            IL(K)=TEMPI
            BR1=BR
            BZ1=BZ
            CALL BPSIC0(ZM,RM,CR,CZ,CI,NC,K)
            BR1=BR+BR1
            BZ1=BZ+BZ1
            IF(IFL1.EQ.1)GOTO 45
            CALL BPSIC0(ZM,RM,ACR,ACZ,ACI,NAC,K)
            BR1=BR1+BR
            BZ1=BZ1+BZ
45          BT=2.*CURZ/RM
            P(K)=P(K)+(BR1**2+BZ1**2+BT**2)/(8*3.14159)
            DLS(K)=SQRT((RL(K1)-RM)**2+(ZL(K1)-ZM)**2)
50          CONTINUE
          IF(IFG.EQ.2)GOTO 100
          IF(IFG.NE.NFG)GOTO 200
C
```

```
C--------PRINT OUT PLASMA MODEL AND PRESSURE
1[illegible]      TYPE 110,IFL,INAME,IPNAME,ICNAME,CURZ
110      FORMAT(1X,1A1,'FILE=',A5/' PLASMA=',A5/' CCOND=',A5
         1 /' Z CURRENT=',E9.3)
           DO 120 I=1,(NL+1)/2
           TYPE 125,I,RL(I),ZL(I),IL(I),P(I),LLF(I)
120        CONTINUE
125      FORMAT(1X,'J=',I3,' R=',F6.3,' Z=',F6.3,' I=',E9.3,
         1 ' P=',F9.3,' D=',F5.2)
         IF(M1.EQ.4)RETURN
C
C--------INPUT CHANGES IN PLASMA POSITION
130      TYPE 140,NL
140      FORMAT(' NO CHANGE FOR,CHANGE:INPUT # OF COILS (1-',I3,')',
         1 ' AUTO CHANGE: INPUT 1 ',$)
         ACCEPT 145,NC
145      FORMAT(I3)
         IF(NC.EQ.0)GOTO 200
         IF(NC.EQ.1)GOTO 200
         [illegible]
         [illegible]
150      TYPE 155
155      FORMAT(1X,'INPUT CHANGES:J,R(J),Z(J)   (EG:5,14.02,0.0)'/
         1 ' WHEN CHANGES INPUT"-1"',)
160      ACCEPT 165,J,[illegible]
165      FORMAT(I3,2F10.3)
         IF(J.EQ.-1)GOTO 1[illegible]
         [illegible]
         [illegible]
         JJ=NL+1-I
         IF(JJ.EQ.NL+1)GOTO 1[illegible]
         [illegible]
         [illegible]
         [illegible]
1[illegible]      [illegible]

C--------[illegible]
[illegible]
```

```
C--------OUTPUT TO DISC THE NEW PLASMA
         CALL ASSIGN(1,IPLAME,16)
         WRITE(1,212)NL
212      FORMAT(I3)
         WRITE(1,215)(RL(I).ZL(I),I=1,NL)
         CALL CLOSE(1)
215      FORMAT(2E10.2)
C
C--------CONTINUE WITH PLASMA CHANGES
         IFG=0
220      IFG=IFG-1
C
C--------FIND NORMALS AND DISTANCES
           DO 250 K=1,NL
           K1=K+1
           K2=K-1
           IF(K.EQ.1)K2=NL
           IF(K.EQ.NL)K1=1
245        RN1(K)=-(ZL(K1)-ZL(K2))
           ZN1(K)=-(RL(K2)-RL(K1))
248        RMOD1=SQRT(RN1(K)**2+ZN1(K)**2)
           RN1(K)=RN1(K)/RMOD1
           ZN1(K)=ZN1(K)/RMOD1
250        CONTINUE
C
C--------MOVE PLASMA
           DO 260 K=1,NL
           RL(K)=RL(K)+(P0-P(K))*RN1(K)*AISTEP*2.5E-9
           IF(K.EQ.1)GOTO 260
           ZL(K)=ZL(K)+(P0-P(K))*ZN1(K)*AISTEP*2.5E-9
260        CONTINUE
C
C--------SYMMETRIZE PLASMA ABOUT Z AXIS
           DO 270 I=2,(NL+1)/2
           ZL(I)=(ZL(I)-ZL(NL+2-I))/2.
           ZL(NL+2-I)=-ZL(I)
           RL(I)=(RL(I)+RL(NL+2-I))/2.
           RL(NL+2-I)=RL(I)
270        CONTINUE
C
C--------BRANCH TO CALCULATE CURRENTS
         GOTO 20
500      CALL COIL(2.IPLAME,RL,ZL,IL,NL)
         RETURN
         END
C
C
```

```
C-------SUBROUTINE TO GET BR AND PSI AT Z,R
        SUBROUTINE BPSICO(Z1,R1,RL,ZL,IL,NL,JJ)
        REAL IL(NL),RL(NL),ZL(NL)
        REAL*8 AM(30,30),Y(30)
        COMMON/BLKB/BR,BZ
        COMMON/BLKA/AM,Y


C
C--------CALCULATE BR,BZ AND PSI
        BZ=0.
        BR=0.
        Y(JJ)=0.
        Z=Z1+.000001
        R=R1+.000001
          DO 100 L=1,NL
          IF(IL(L))5,100,5
5         AZ=Z-ZL(L)
          CAY2=(4.*RL(L)*R)/((RL(L)+R)**2+AZ**2)
          PF=(.5*(2-CAY2)*ELK(CAY2)-ELE(CAY2))/SQRT(CAY2)
          D=SQRT((RL(L)+R)**2+AZ**2)
          D1=(RL(L)-R)**2+AZ**2
          BR=BR+2.*IL(L)*AZ/(R*D)*(-ELK(CAY2)+(RL(L)**2+R**2+AZ**2)
        1  *ELE(CAY2)/D1)
          BZ=BZ+2.*IL(L)/D*(ELK(CAY2)+(RL(L)**2-R**2-AZ**2)
        1  *ELE(CAY2)/D1)
          Y(JJ)=Y(JJ)+IL(L)*PF*4.*SQRT(R*RL(L))
100       CONTINUE
        Y(JJ)=-Y(JJ)
        RETURN
        END
```

```
C       ***** EPSICAL.FTN *****
C
C--------SUBPROGRAM TO CALCULATE VALUES OF PSI IN VACUUM
C--------FOR AN ARBITRARY SET OF CURRENT LOOPS
C--------THIS PROGRAM CALCULATES PSI AT 0.5 GRID INTERVALS
        SUBROUTINE PSICAL(AR,AZ,BI,NA)
        REAL AR(NA),AZ(NA),BI(NA)
        COMMON/BLK6/RSTEP,ZSTEP,IRSTRT,IZSTRT
        COMMON/BLKC/PSI(43,41),PH,PL
C
C--------CALCULATE GRID OF PSI VALUES
302     PL=1.E30
        PH=-1.E30
          DO 110 I1=1,43
          DO 110 J1=1,41
          AJ=(J1-1)*RSTEP+IRSTRT+.000001
          AI=(I1-3.)*ZSTEP+IZSTRT+.000001
            DO 100 L=1,NA
            D=(AR(L)+AJ)**2+(AI-AZ(L))**2
              CAY2=4.*AR(L)*AJ/D
              PF=(.5*(2.-CAY2)*ELK(CAY2)-ELE(CAY2))/SQRT(CAY2)
100           PSI(I1,J1)=PSI(I1,J1)+BI(L)*PF*4*SQRT(AJ*AR(L))
            PH=AMAX1(PSI(I1,J1),PH)
110       PL=AMIN1(PSI(I1,J1),PL)
        RETURN
        END
```

```
C         ***** ECURR.FTN *****
C
C--------SUBPROGRAM TO CALCULATE CURRENTS INDUCED IN A SET OF
C--------TOROIDALLY SYMMETRIC CONDUCTORS.
C--------USES CHOLESKY DECOMPOSITION FOR AM*X=Y,AM=AL*D*AL', WHERE
C--------AM IS THE INDUCTANCE MATRIX, X IS THE CURRENT VECTOR, AND Y IS
C--------THE PSI VECTOR.
C--------REF. KERSHAW,'J. COMP. PHYS.', 26,43,(1978)
      SUBROUTINE CURR(UR,UZ,UI,NU,LU,U0)
      REAL UR(NU),UZ(NU),UI(NU)
      REAL*8 AM(30,30),AL(30,30),X(30),Y(30)
      COMMON/BLKA/AM,Y
      COMMON/BLKB/CR(50),CZ(50),CI(50),NC
C--------GET INDUCTANCES
        DO 10 I=1,NU
        DO 10 J=1,NU
        CALL LMAT(I,J,UR,UZ,NU)
10      CONTINUE
C
C--------GET PSI'S FOR PLASMA AND...
        DO 20 II=1,LU
        CALL BPSICO(UZ(II),UR(II),CR,CZ,CI,NC,II)
        Y(II)=Y(II)+U0
20      CONTINUE
      IF(LU.EQ.NU)GOTO 25
C
C--------...FOR CENTER CONDUCTOR
        DO 25 II=LU+1,NU
        CALL BPSICO(UZ(II),UR(II),CR,CZ,CI,NC,II)
25      CONTINUE
C
C--------NORMALIZE THE INDUCTANCE MATRIX AND PSI'S
      CHAMP=0.
        DO 30 I=1,NU
        DO 30 J=1,NU
        TEM=ABS(AM(I,J))
        IF(TEM.GT.CHAMP)CHAMP=TEM
30      CONTINUE
C
        DO 40 I=1,NU
        Y(I)=Y(I)/CHAMP
          DO 40 J=1,NU
          AM(I,J)=AM(I,J)/CHAMP
40      CONTINUE
C
```

```
C--------GET AL
          DO 50 I=1,NU
          DO 50 J=I,NU
C-----  NOTE I, NOT 1
          AL(J,I)=AM(J,I)
          IF(I.EQ.1)GOTO 50
          SUM=0.
            DO 45 K=1,I-1
45          SUM=SUM+AL(J,K)*AL(I,K)/AL(K,K)
C-----  NEED NOT REFER TO D EXPLICITLY, SINCE D(I)=1/AL(I,I)
          AL(J,I)=AM(J,I)-SUM
50        CONTINUE
C
C--------START BACK SUBSTITUTION
        X(1)=Y(1)/AL(1,1)
          DO 70 MI=2,NU
          SUM=0.
            DO 60 NI=1,MI-1
60          SUM=SUM+X(NI)*AL(MI,NI)


          X(MI)=(Y(MI)-SUM)/AL(MI,MI)
70        CONTINUE
C       CONTINUE BACK SUBSTITUTION
          DO 80 MI=1,NU
          X(MI)=X(MI)*AL(MI,MI)
80        CONTINUE
C
        X(NU)=X(NU)/AL(NU,NU)
          DO 100 MI=NU-1,1,-1
          SUM=0.
            DO 90 NI=MI+1,NU
90          SUM=SUM+X(NI)*AL(NI,MI)
          X(MI)=(X(MI)-SUM)/AL(MI,MI)
100       CONTINUE
C
          DO 110 I=1,NU
          UI(I)=X(I)
110       CONTINUE
        RETURN
        END
```

```
C
C--------SUBPROGRAM TO GET THE INDUCTANCE MATRIX
         SUBROUTINE LMAT(I,J,R,Z,N)
         REAL R(N),Z(N)
         REAL*8 AM(30,30),Y(30)
         COMMON/BLKA/AM,Y
C
C--------CALCULATE INDUCTANCE
         ZI=Z(I)+.000001
         RI=R(I)+.000001
         AZ=ZI-Z(J)
         Q=(R(J)+RI)**2+AZ**2
         CAY2=(4.*R(J)*RI)/Q
         PF=(.5*(2.-CAY2)*ELK(CAY2)-ELE(CAY2))/SQRT(CAY2)
         AM(I,J)=PF*4.*SQRT(RI*R(J))
         RETURN
         END
```

```
C         ***** BDRAW.FTN *****
C
C--------SUBPROGRAM TO DISPLAY PSI AND MOD B
C--------THIS PROGRAM ALSO PLOTS B VS R, AND B VS Z
C--------THE GRID IN THIS VERSION IS 40 X 43
C--------THIS PROGRAM IS TO BE USED IN CONJUNCTION WITH
C--------THE MAIN PROGRAM EPLAPO, AND WITH EPSICAL, BBEW, SPLIN, ECURR
C--------AND IS TO BE LINKED WITH GRAF AND CONTURE
C--------THE PSI AND B VALUES ARE CALCULATED EVERY GRID POINT
C
        SUBROUTINE DRAW(M3)
C--     M3=0 > PLASMA
C--     M3=1 > VACUUM
C
        REAL IL(50),FNZ(43),E(43,41)
        COMMON/BLKC/PSI(43,41),PH,PL
        COMMON/BLKD/B(43,41),BH,BL
        COMMON/BLK1/RL(50),ZL(50),IL,NL,IPLAME(8),PSI0
        COMMON/BLK2/CR(50),CZ(50),CI(50),NC,IVAME(8)
        COMMON/BLK3/ACR(50),ACZ(50),ACI(50),NAC,ICNAME(8)
        COMMON/BLK5/IPSAME(8),CURZ,IFLG,JFLG
        COMMON/BLK6/RSTEP,ZSTEP,IRSTRT,IZSTRT
        DATA MP,MB,MX,MG,MZ,PI,NO/2HP ,2HB ,2HEX,2HBR,2HBZ,3.14159,2HNO/
        IGS=31
        IBL=7
C
C--------CALCULATE PSI AND B MATRICES
          DO 305 I=1,43
          DO 305 J=1,41
305       PSI(I,J)=0
C
C----    IF NO VACUUM PSI FILE GO TO 310
        IF(JFLG.EQ.0)GOTO 310
C
C----    INPUT VACUUM PSI FILE
        CALL ASSIGN(1,IPSAME,16)
        READ(1,ERR=306)PSI,PL,PH
        CALL CLOSE(1)
        GOTO 320
306     RETURN
C
C----    CALCULATE VACUUM PSI, AND OUTPUT FILE
310     CALL PSICAL(CR,CZ,CI,NC)
311     TYPE 312,IGS,IBL
312     FORMAT(1X,2A1,'OUTPUT PSI FILENAME?(OR NO) ',$)
        READ(5,314,ERR=311)IPSAME
314     FORMAT(8A2)
        IF(IPSAME(1).EQ.NO)GOTO 320
        CALL ASSIGN(1,IPSAME,16)
        WRITE(1)PSI,PL,PH
        CALL CLOSE(1)
        JFLG=1
```

```
C
C----    IF NO CENTER CONDUCTOR GO TO 330
320      IF(IFLG.EQ.1)GOTO 330
         CALL PSICAL(ACR,ACZ,ACI,NAC)
C
C----    FOR VACUUM PLOT GO TO 350
330      IF(MZ.EQ.1)GOTO 350
C
C----    CALCULATE PSI FROM PLASMA
         CALL PSICAL(RL,ZL,IL,NL)
         CALL BER(0)
         GOTO 400


350      CALL BER(1)
C
C--------CALCULATE THE TOTAL ENERGY OVER THE GRID
400        DO 110 J=1,41
110        ENZ(J)=0
         ENG=0
           DO 150 I=3,43
             DO 160 J=2,
160          ENZ(I)=ENZ(I)+B(I,J)**2*((J-1)*RSTEP+IRSTRT)*RSTEP
           ENZ(I)=ENZ(I)+B(I,1)**2*IRSTRT*RSTEP/2.
           ENZ(I)=ENZ(I)+B(I,41)**2*(40.*RSTEP+IRSTRT)*RSTEP/2.
           IF(I.EQ.3)GOTO 148
           IF(I.EQ.43)GOTO 148
           ENG=ENG+ENZ(I)*ZSTEP
           GOTO 150
148        ENG=ENG+ENZ(I)*ZSTEP/2.
150        CONTINUE
         ENG=ENG/[illegible]
C
C--------PROVIDE MISCELLANEOUS DISPLAYS
500      CALL TYPH(2,1,0)
510      TYPE 7,IGS,III
7        FORMAT(1X,2A1,'DISPLAY: P=PSI,B=B,EX=EXIT,BR=B VS
         1 [illegible]
         [illegible]
         [illegible]
         CALL [illegible]
         [illegible]
         CALL [illegible]
         [illegible]
         FORMAT([illegible]
         [illegible]
C
C--------BRANCH FOR THE APPROPRIATE DISPLAY
         IF(ICMD.EQ.'P ') GO TO [illegible]
         IF(ICMD.EQ.'EX') GO TO [illegible]
C
```

```
C---      PSI DISPLAY
30        TYPE 32,PL,PH,BL,BH,NOMD
32        FORMAT(' PSILO,PSIHI'/2E11.3/' BLO,BHI'/2E11.3
          1  /' ENTER ',1A2,': N.LO,HI'/' (EG:20,-1.E6,1.E6)'
          2  /' N=EVEN # OF CONTOURS'/' BETWEEN LO & HI')
          READ(5,11,ERR=30)NA,AL,AH
          AN=NA
11        FORMAT(I3,2E11.3)
          ADOT=(AH+AL)/2
          AGAP=(AH-AL)/AN
          TYPE 12,ADOT,AGAP,IVAME,IPLAME,ICOAME,CURZ,ENG,PSI)
12        FORMAT(' DOT=',E11.3/' SPACE=',E11.3/' FILE=',8A2/' PLASMA=',8A2
          1 /' COOND=',8A2/' Z CURRENT=',E9.3/' ENERGY=',E11.3
          2 /' PSI0=',E11.3)
          N=AN
          CALL GRID(3.,43.,300,1000,20,1.,41.,50,750,20)
            DO 50 I=40,740,174
            CALL TVPU(1000,I)
            J=10*(I-40)/174*RSTEP+IRSTRT
            TYPE 55,IGS,J
55          FORMAT(1X,1A1,I2)
50          CONTINUE
C
            DO 59 I=275,995,174
            CALL TVPU(I,25)
            J=10*(I-275)/174*ZSTEP-IZSTRT
            TYPE 57,IGS,J


57          FORMAT(1X,1A1,I3)
59          CONTINUE
C
C--------DRAW "X'S" REPRESENTING PLASMA BORDER
          IF(M3.EQ.1)GOTO 66
            DO 65 I=1,NI
            BZ=(ZL(I)-IZSTRT)/ZSTEP+3
            BR=(RL(I)-IRSTRT)/RSTEP+1
            IF(BR.LT.1.OR.BR.GT.41)GOTO 65
            IF(BZ.LT.3.OR.BZ.GT.43)GOTO 65
            CALL PLOT(BZ,BR,1,'X')
65          CONTINUE
66        CONTINUE
            DO 69 I=1,NC
            BZ=(CZ(I)-IZSTRT)/ZSTEP+3
            BR=(CR(I)-IRSTRT)/RSTEP+1
            IF(BR.LT.1.OR.BR.GT.41)GOTO 69
            IF(BZ.LT.3.OR.BZ.GT.43)GOTO 69
            CALL PLOT(BZ,BR,1,'Y')
69          CONTINUE
          IF(NOMD.EQ.MP)CALL CONTUR(PSI,43,41,AL,AH,N)
          IF(NOMD.EQ.MB)CALL CONTUR(B,43,41,AL,AH,N)
          GOTO 200
```

```
C
C-------GRAPH OF B VS R
72      TYPE 73,BL,BH
73      FORMAT(/' BLO,BHI'/2E11.3/' ENTER BHI,Z'/' (EG:3.E4,3.) ')
        READ(5,13,ERR=70)AH,Z
13      FORMAT(E11.3,F6.2)
        TYPE 74,CURRZ,IVAME,IPLAME,PSI0,Z
74      FORMAT(' Z CURRENT=',E9.3/' FILE=',8A2///' PLASMA=',8A2/
        1' PSI0=',E9.3///' B VS R'//' Z=',F6.2)
        START=IRSTRT
        ASTOP=IRSTRT+40.*RSTEP
        CALL GRID(START,ASTOP,300,1000,20,0.,AH,50,750,20)
          DO 99 I=290,995,174
          CALL TVPU(I,25)
          J=10*(I-290)/174*RSTEP+IRSTRT
99        TYPE 55,IGS,J
C
          DO 95 I=40,740,174
          CALL TVPU(153,I)
          J=(I-40)/174
          AJ=(AH)/4.*J
          TYPE 56,IGS,AJ
56        FORMAT(1X,1A1,F11.3)
95        CONTINUE
        IZ=(Z-IZSTRT)/ZSTEP+3
        CALL PLOT(START,B(IZ,1),1,0)
          DO 78 IR=1,41
          R=(IR-1)*RSTEP+IRSTRT
78        CALL PLOT(R,B(IZ,IR),1,1)
        GOTO 200
C
```

```
C--------GRAPH OF B VS Z
80        TYPE 82,BL,BH
82        FORMAT(/' BLO,BHI'/2E11.3/' ENTER BHI,R'/' (EG:3.E4,10.) ')
          READ(5,13,ERR=80)AH,R
83        TYPE 84,CURZ,IVAME,IPLAME,PSI0
84        FORMAT(' Z CURRENT=',F9.3/' FILE=',6A2///' PLASMA=',6A2
          1/' PSI0=',F9.3)
          START=IZSTRT
          RSTOP=IZSTRT+40.*ZSTEP
          TYPE 502,R


502       FORMAT(1X///' B VS Z'///' R=',F6.2)
          CALL GRID(START,RSTOP,200,1000,20,0.,AH,50,750,20)
142         DO 86 I=275,995,174
            CALL TVPU(I,25)
            J=10*(I-275)/174*ZSTEP+IZSTRT
86          TYPE 87,IGS,J
C
            DO 88 I=40,740,174
            CALL TVPU(150,I)
            J=(I-40)/174
            AJ=AH/4.*J
            IF(KFLG.EQ.1)AJ=(AH-AL)/4.*J+AL
            TYPE 86,IGS,AJ
88          CONTINUE
          IR=(R-IRSTRT)/RSTEP+1
          CALL PLOT(0.,B(3,IR),1,0)
            DO 89 IZ=3,43
            Z=(IZ-3.)*ZSTEP+IZSTRT
89          CALL PLOT(Z,B(IZ,IR),1,1)
          GO TO 200
          END
```

```
C          ***** GRAF.FTN *****
C---THIS IS A SET OF ROUTINES FOR MAPPING AND PLOTTING
C---DATA ON THE TEKTRONIX 4006 WHICH CAN BE CALLED
C---FROM FORTRAN PROGRAMS.
C                   JOHN COONROD     SEPTEMBER 1977
        SUBROUTINE PLOT(X,Y,N,M)
C---SUBROUTINE PLOT PLOTS POINT(S) ACCORDING TO A
C---MAPPING FROM A PRIOR CALL TO GRID
        REAL X(N),Y(N)
        LOGICAL LF,XLOG,YLOG
        LOGICAL*1 AMSG(2)
        COMMON /GP1977/ XLOG,YLOG,XLO,YLO,DX,DY,SX,SY,IXL,IYL
        DATA XLOG,YLOG.IXL.IYL.SX,SY/.FALSE.,.FALSE.,0,0,1.,1./
        AMSG(1)="237
        AMSG(2)=M
        DO 1 I=1,N
        IF(XLOG)IX=IXL+DX*ALOG10(X(I)/XLO)
        IF(.NOT.XLOG)IX=IXL+(X(I)-XLO)*SX
        IF(YLOG)IY=IYL+DY*ALOG10(Y(I)/YLO)
        IF(.NOT.YLOG)IY=IYL+(Y(I)-YLO)*SY
        IF(M.LT.2)GO TO 10
        CALL TVPU(IX-4,IY-4)
        CALL TO IO(AMSG,2)
        GOTO 1
10      LF=((M.EQ.0).OR.((M.GT.1).AND.(I.EQ.1)))
        IF(LF) CALL TVPU(IX,IY)
        IF(.NOT.LF) CALL TVPD(IX,IY)
1       CONTINUE
        RETURN
        END
        SUBROUTINE GRID(X1,X2,IX1,IX2,N,Y1,Y2,IY1,IY2,M)
C---THIS SUBROUTINE DEFINES A LINEAR OR LOG MAPPING
C---AND OUTLINES IT IN A GRID.
C---      N=0 --- MAKE    LINEAR GRID MARKS TWIXT X1 AND X2
C---      N>0 --- MAKE N LOG. MARKS PER DECADE, (ISKIP DEFINE
C---                 HOW MANY DECADES
C---      N<0 --- NO MARKS, LINEAR SCALE
C---SIMILARLY WITH M FOR THE Y AXIS.  THE INTEGER LIMITS
C---DEFINE THE CORNERS OF THE GRID ON THE 1023X780 SCREEN.
        LOGICAL XLOG,YLOG,X1LOG,Y1LOG
        COMMON /GP1977/ XLOG,YLOG,XLO,YLO,DX,DY,SX,SY,IXL,IYL
        CALL ERASEP(67,,,,.FALSE.,)
        IXL=IX1
        IYL=IY1
        XLO=X1
        YLO=Y1
        XLOG=N.LT.0
        YLOG=M.LT.0
        X1LOG=N.EQ.0
        Y1LOG=M.EQ.0
```

```
          SX=(IX2-IX1)/(X2-X1)
          IF(X1LOG) GO TO 30
          IF(XLOG) GO TO 20
          DO 10 I=0,N
          IX=IX1+I*(IX2-IX1)/N
          CALL TVPU(IX,IY1)
 10       CALL TVPD(IX,IY2)
          GOTO 30
 20       SX=X1/X2
          NI=-N
          NJ=-ALOG10(SX)
          XNJ=NJ
          IF(XNJ.NE.(-ALOG10(SX)))NJ=NJ+1
          DX=(IX2-IX1)/NJ


          CALL TVPU(IX1,IY1)
          CALL TVPD(IX1,IY2)
          DO 22 I=1,NI
          AX=ALOG10(I*10./NI)
          DO 22 J=1,NJ
          IY=IX1+DY*(J-1+AX)
          CALL TVPU(IX,IY1)
 22       CALL TVPD(IX,IY2)
 30       SY=(IY1-IY2)/(Y1-Y2)
          IF(Y1LOG) GO TO 100
          IF(YLOG) GO TO 40
          DO 32 I=0,N
          IY=IY1+I*(IY2-IY1)/N
          CALL TVPU(IX1,IY)
 32       CALL TVPD(IX2,IY)
          GOTO 100
 40       SY=Y1/Y2
          NJ=-ALOG10(SY)
          XNJ=NJ
          IF(XNJ.NE.(-ALOG10(SY)))NJ=NJ+1
          DY=(IY2-IY1)/NJ
          NI=-N
          CALL TVPU(IX1,IY1)
          CALL TVPD(IX2,IY1)
          DO 42 I=1,NI
          AY=ALOG10(I*10./NI)
          DO 42 J=1,NJ
          IY=IY1+DY*(J-1+AY)
          CALL TVPU(IX1,IY)
 42       CALL TVPD(IX2,IY)
 100      RETURN
          END
```

```
SUBROUTINE TVCL
LOGICAL*1 A(2)
A(1)="233
A(2)="214
CALL DOQIO(A,2)
CALL MARK(2,45,1)
CALL WAITFR (2)
RETURN
END
SUBROUTINE DOQIO(A,N)
INTEGER IPAR(6),ISB(2)
CALL GETADR(IPAR(1),A)
IPAR(2)=N
CALL QIO("410,5,1,,ISB,IPAR)
CALL WAITFR(1)
RETURN
END
SUBROUTINE TVPU(IX,IY)
CALL TVPO(IX,IY,1)
RETURN
END
SUBROUTINE TVPD(IX,IY)
CALL TVPO(IX,IY,2)
RETURN
END
SUBROUTINE TVPO(IX,IY,I1)
LOGICAL*1 A(5)
I2=6-I1
IF(IX.LT.0)IX=0
IF(IX.GT.1023)IX=1023
IF(IY.LT.0)IY=0
IF(IY.GT.1023)IY=1023


A(1)="2[illegible]5
A(2)="2[illegible]0.OR.(IY/32)
A(3)="34[illegible].OR.(IY.AND.31)
A(4)="1[illegible]3.OR.(IX/32)
A(5)="3[illegible]0.OR.(IX.AND.31)
CALL DOQIO(A(I1),I2)
RETURN
END
```

```
C       ***** CONTURE.FTN *****
C
C---CONTUR PLOTS NC LINEARLY SPACED CONTOURS FOR ARRAY
C---A(NX,NY) BETWEEN THE VALUES AL AND AH.
C---GRID MUST BE CALLED FIRST TO SCALE THE MAP TO NX BY NY
C---THE PROGRAM GOES THROUGH THE ARRAY 4 POINTS
C---(1 SQUARE) AT A TIME.  IT REMAPS THE 4 POINTS SO THE
C---RANGE AL TO AH FALLS BETWEEN 1 AND NC.  IT CHECKS TO
C---SEE WHICH CONTOURS CROSS THE SQUARE.  FOR EACH CROSSING
C---CONTOUR IT CALCULATES THE POINTS ON THE BOUNDARY IT
C---CROSSES, AND JOINS THOSE POINTS WITH STRAIGHT LINES.
C---                JOHN COONROD    SEPTEMBER 1977
      SUBROUTINE CONTUR(A,NX,NY,AL,AH,NC)
      REAL A(NX,NY),B(5),Z(4)
      INTEGER IB(4)
      LOGICAL L(4),LDOT
      D=NC/(AH-AL)
      IMIDC=NC/2
      A0="237
      DO 4 I=4,NX
      B(1)=D*(A(I,1)-AL)
      B(4)=D*(A(I-1,1)-AL)
      DO  4 J=2,NY
      B(2)=B(1)
      B(3)=B(1)
      B(1)=D*(A(I,J)-AL)
      B(4)=D*(A(I-1,J)-AL)
      B(5)=B(1)
      IC1=-10000
      IC2=10000
      DO 1 K=1,4
      IB(K)=B(K)
      IF(IB(K).GT.IC1)IC1=IB(K)
1     IF(IB(K).LT.IC2)IC2=IB(K)
      IF(IC1.GT.NC) IC1=NC
      IF(IC1.LE.IC2) GO TO 4
      IC3=IC2+1
      IF(IC3.LE.0) IC3=1
      IF((IC1-IC3).GT.NC)GOTO 999
```

```
          DO 3 K=IC3,IC1
          LDOT=(K.EQ.IMIDC)
          DO 2 LL=1,4
          DZ=(B(LL)-B(LL+1))
          IF(ABS(DZ).LE.1.E-3)DZ=1.E-3
          Z(LL)=(B(LL)-K)/DZ
2         L(LL)=(Z(LL).GE.0.).AND.(Z(LL).LE.1.)
          Q=I
          R=J
          IF(L(1).AND.L(2)) CALL VEC(LDOT,Q,R-Z(1),Q-Z(2),R-1)
          IF(L(1).AND.L(3)) CALL VEC(LDOT,Q,R-Z(1),Q-1,R-1+Z(3))
          IF(L(1).AND.L(4)) CALL VEC(LDOT,Q,J-Z(1),Q-1+Z(4),R)
          IF(L(2).AND.L(3)) CALL VEC(LDOT,Q-Z(2),R-1,Q-1,R-1+Z(3))
          IF(L(2).AND.L(4)) CALL VEC(LDOT,Q-Z(2),R-1,Q-1+Z(4),R)
3         IF(L(3).AND.L(4)) CALL VEC(LDOT,Q-1,R-1+Z(3),Q-1+Z(4),R)
4         CONTINUE
          RETURN
999       IGS="237
          CALL TYPE(2,1000)
          TYPE 1999,IGS,IC3,IC1
1999      FORMAT(1X,1A1,2I6)
          RETURN
          END
C---SUBROUTINE VEC PLOTS A VECTOR IN A MAPPED SPACE
          SUBROUTINE VEC(LDOT,X1,Y1,X2,Y2)

          CALL PLOT (X1,Y1,1,0)
          IF(.NOT.LDOT)CALL PLOT(X2,Y2,1,1)
          IF(.NOT.LDOT)RETURN
          XM=(X2+X1)/2
          YM=(Y2+Y1)/2
          CALL PLOT(XM,YM,1,1)
          RETURN
          END
```

```
C         ***** BBFN.FTN *****
C
C-------- SUBPROGRAM TO CALCULATE B AT GRID POINTS
          SUBROUTINE BFN(MVAC)
          REAL IL(50),DL(50)
          COMMON/BLK0/PSI(43,41),PH,PL
          COMMON/BLK2/B(43,41),BH,BL
          COMMON/BLK1/RL(50),ZL(50),IL,NL,IPLAME(8),PSI0
          COMMON/BLK5/IPSAME(8),CURZ,IFLG,JFLG
          COMMON/BLK6/RSTEP,ZSTEP,IRSTRT,IZSTRT
          COMMON/BLK7/R(50),RN(50),ZN(50),DL5(50),R1
C
C---------CALCULATE THE B ARRAY
          BL=1.E33
          BH=-BL
            DO 105 J1=3,39
            AJ=(J1-1)*RSTEP+IRSTRT
            BT=2.*CURZ/(AJ+.000001)
              DO 100 I=3,41
              BR=(PSI(I+2,J1)-PSI(I-2,J1)-8*(PSI(I+1,J1)-PSI(I-1,J1)))/
          1   (12*ZSTEP*AJ)
              BZ=(8*(PSI(I,J1+1)-PSI(I,J1-1))-(PSI(I,J1+2)-PSI(I,J1-2)))/
          1   (12*RSTEP*AJ)
              B(I,J1)=SQRT(BT*BT+BR*BR+BZ*BZ)
              BH=AMAX1(BH,B(I,J1))
100           BL=AMIN1(BL,B(I,J1))
105         CONTINUE
C
            DO 110 I=3,41
            B(I,2)=2*B(I,3)-B(I,4)
            B(I,1)=2*B(I,2)-B(I,3)
            B(I,40)=2*B(I,39)-B(I,38)
            B(I,41)=2*B(I,40)-B(I,39)
110         CONTINUE
C
            DO 115 I=1,41
            B(2,I)=2*B(3,I)-B(4,I)
            B(1,I)=2*B(2,I)-B(3,I)
            B(42,I)=2*B(41,I)-B(40,I)
            B(43,I)=2*B(42,I)-B(41,I)
115         CONTINUE
200       RETURN
          END
```

APPENDIX II

USERS GUIDE

This guide is designed to enable a person with no programming experience to use the code. It is assumed that the code has been implemented on the minicomputer with a graphics terminal. The code solves for the plasma equilibrium shape given a set of external coils and currents, so the user must have a trial set of coils in mind, as well as a trial plasma shape. Computer files form the basis for most inputs. The code will automatically ask for data and then write the files, asking for file names when appropriate. Data is therefore automatically stored and accessible for new calculation, and can be easily changed using the system editor.

First the computer will ask for the initial and final radii of the grid and the initial and final axial dimensions. These should be typed in, fixed point, separated by commas as in the example provided. Note that the grid is for display purposes only. Coils representing the plasma or external conductors can be placed outside the grid.

The next input is the set of external conductors, "Input center conductor" is printed. Either a file name is input, or the R and Z locations of any closed toroidal conductors. For the Tormac experiments a central cylindrical conductor was represented by a set of coils at one radius, evenly spaced in Z. After the conductors are input the computer will ask for a file name and will then automatically store the coil locations in a file for future use and reference. In a like manner the external coils are input, with the addition of the current magnitude for each coil. The currents should be entered in Abamps. A toroidal field is then input by asking for Z current (Abamps). The magnitude of this axial current determines the toroidal field, in lieu of actual toroidal field coils.

The program will next ask for a PSI filename. This is a file with the ψ values for the external coils (not the plasma or center conductor coils) at each grid point. If no filename is entered the computer will calculate these values before presenting a display, and will ask for a filename to store these values. Therefore the old PSI file can be used as long as the grid and the external coil set remain unchanged.

At this point the user can branch to perform a number of calculations. This branching section is repeated throughout the use of the program, so some of the instructions to be explained will only be of use at other points in the calculation.

The instruction "plasma" allows the input of a trial set of plasma coils. Either a file name is entered or the program asks for a specific set of coils. Two points should be noted. There should be an odd number of coils with the first coil at Z=0. The remainder of the coils should form a mirror image about Z=0, and the coils should be input sequentially following the plasma perimeter. This is needed to correctly calculate the normals and distances between coils. After the coil locations (or file) are input, the value of ψ at the plasma surface is input. The ψ value chosen will have its primary effect on the magnitude of the total plasma current, and secondarily on the plasma shape and position. A value of zero implies that there is no current in the plasma prior to energizing the main external coils. The program will next print a list of the coil locations (R and Z), currents, distances between coils, and the magnetic pressures at all coil locations. Only half the coils are printed; mirror image coils are neglected since symmetry about the midplane (Z=0) is assumed. At this point the operator can go on to display the configuration, or can go back to the branching section.

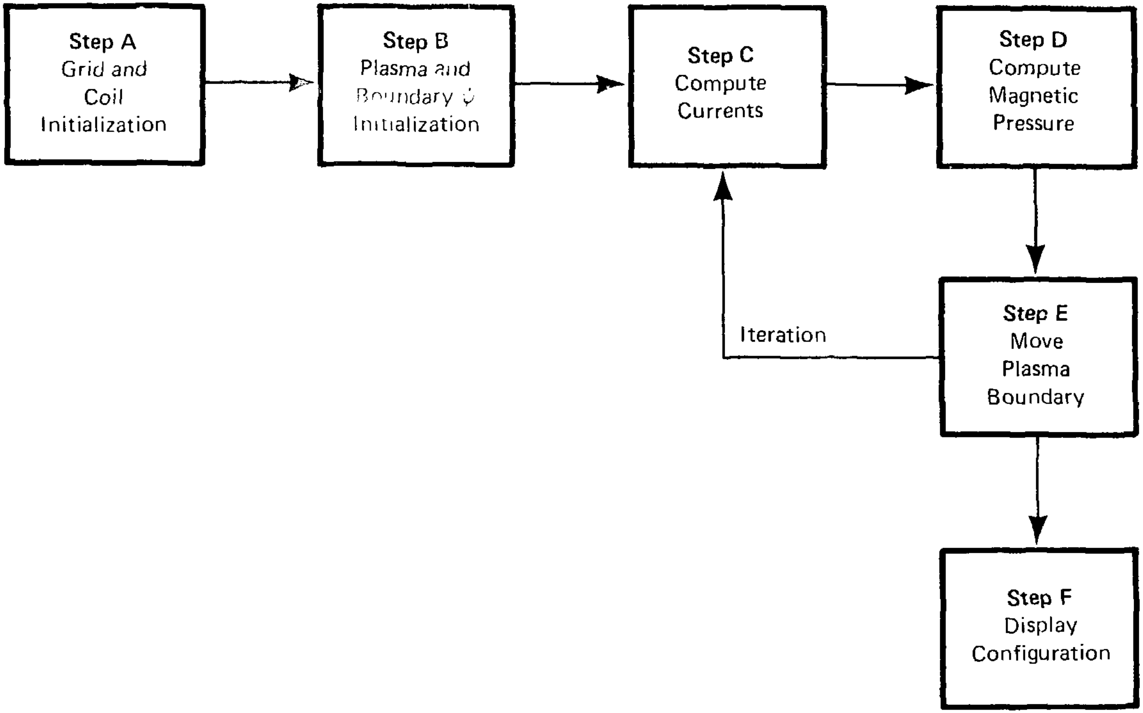
"Change plasma" branches to calculate the pressures on the plasma border, after the trial plasma has been entered. These values are printed out, just as when "plasma" is selected. The program then asks for manual change, where the present number of coils is displayed and the desired number of coils is input, or automatic change, where "1" is input. For a manual change the number of the coil to be changed is input, along with its radial and axial position. The mirror image coil is automatically moved to its new position. This option is typically chosen to even the spacing of coils around the border. If the automatic option is chosen, the program asks for the desired plasma pressure (c.g.s. units) a step size (usually 10. - 100.), and the number of repetitions desired. After the calculation is complete, the new position and pressures are printed. This procedure can be repeated, the new plasma can be stored, and the configuration displayed, or the user can return to the branching section. Note that when automatic changes are selected, the present plasma is automatically stored. This is a precaution in case too large a step size is selected and the iteration diverges.

Other options in the branching section include "New File" which starts the program at the beginning, "New Z Current" which returns the program back to the toroidal field input, and "Coils Only" which calculates the configuration due to the external coils and any fixed closed conductors. Provisions are also made for exiting from the program with "Exit".

The final part of this guide deals with the various displays. Two types of displays are available, contour maps and graphs. The contour maps display either field lines (ψ) or mod $|B|$ ($|B|$) over the entire grid. One inputs the number of lines between two values of psi or field, as well as the two values ("lo" and "hi"). The central contour will be dotted, and the value at the dotted contour as well as the difference between two contours ("dot" and "space") will be printed. The graph display provides a plot of $|B|$ vs R or $|B|$ vs Z. One inputs the maximum value of $|B|$, and the Z or R position, respectively. Note that when either display is requested the maximum and minimum values of the relevant parameter are displayed.

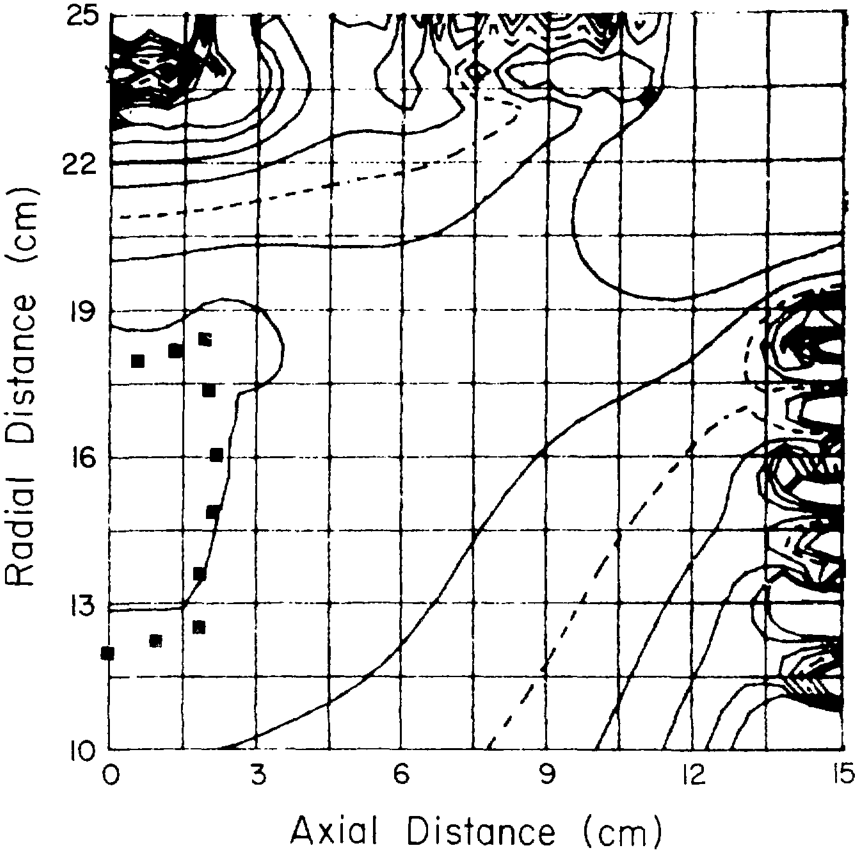
Figure Captions

Fig. 1.	Flow chart of the code. The number of iterations and the step size are user inputs.
Fig. 2.	Initial configurations - $\|B\|$ surfaces a) Elongated trial plasma b) Squat trial plasma
Fig. 3.	Final configurations - $\|B\|$ surfaces a) Convergence of elongated plasma b) Convergence of squat plasma
Fig. 4.	Tormac VB field lines. The squares represent the plasma boundary, while the dark border is the vessel. Note the extensions to allow flow out the cusps.
Fig. 5.	Vacuum field for Tormac VB.
Fig. 6.	Tormac P-1 field lines. The lines can be traced back to the simulated plasma from the particle flux measurement position.
Fig. 7.	Tormac P-1 vacuum field lines. The field lines from the location of the main body of the plasma fall nowhere near the measured particle flux.



XBL 825-555

Figure 1



XBL 824 541

Figure 2a

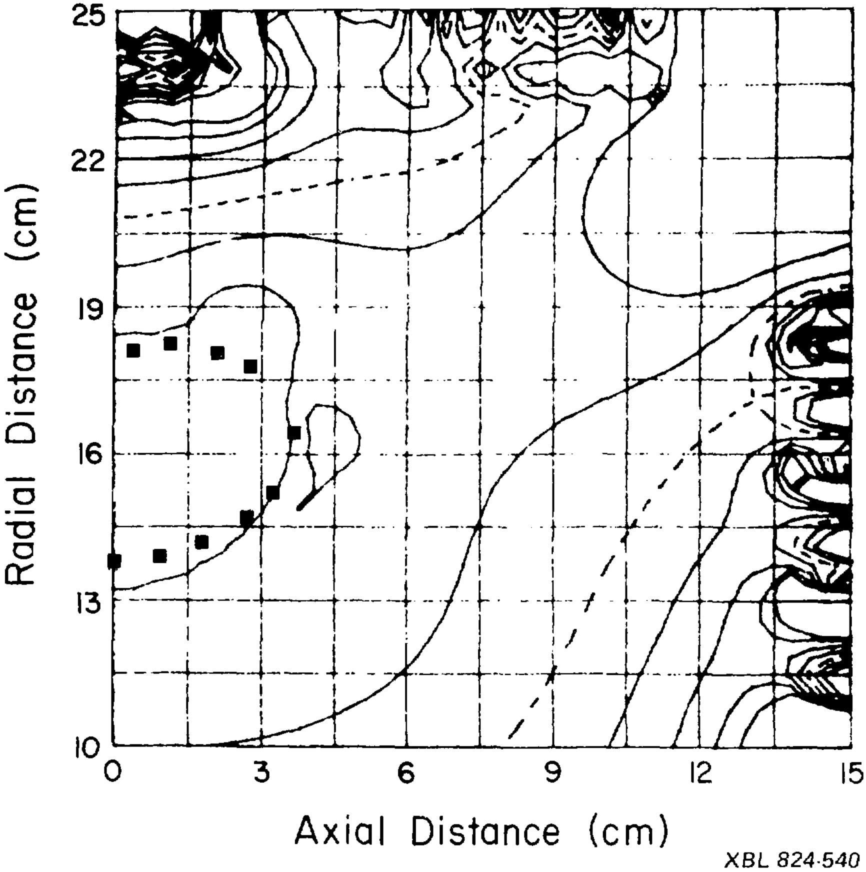
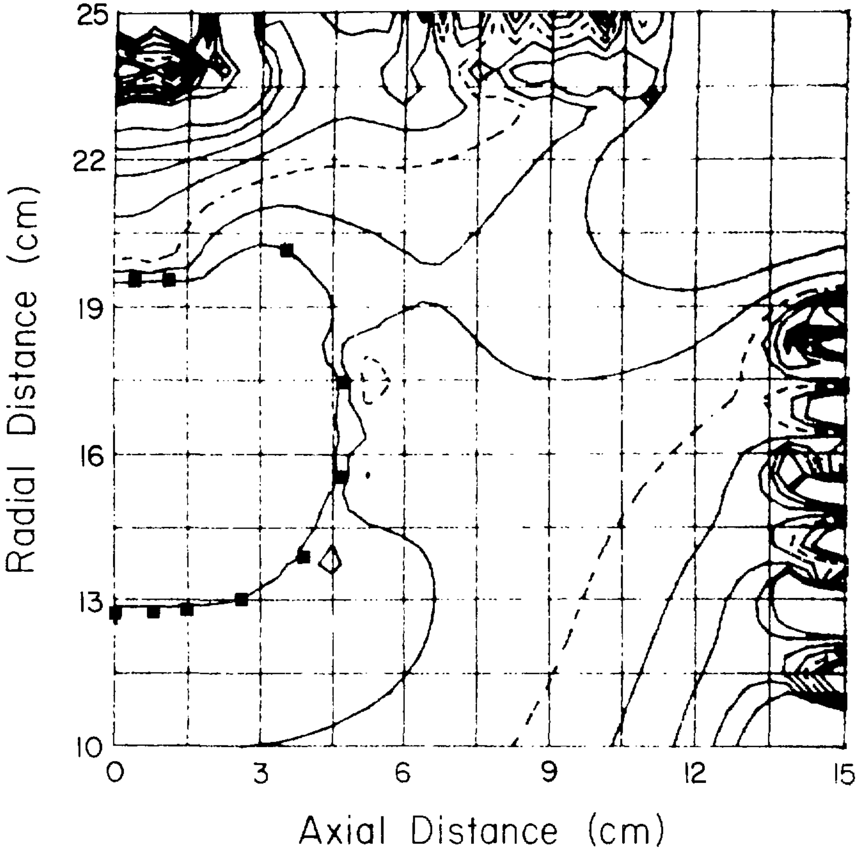
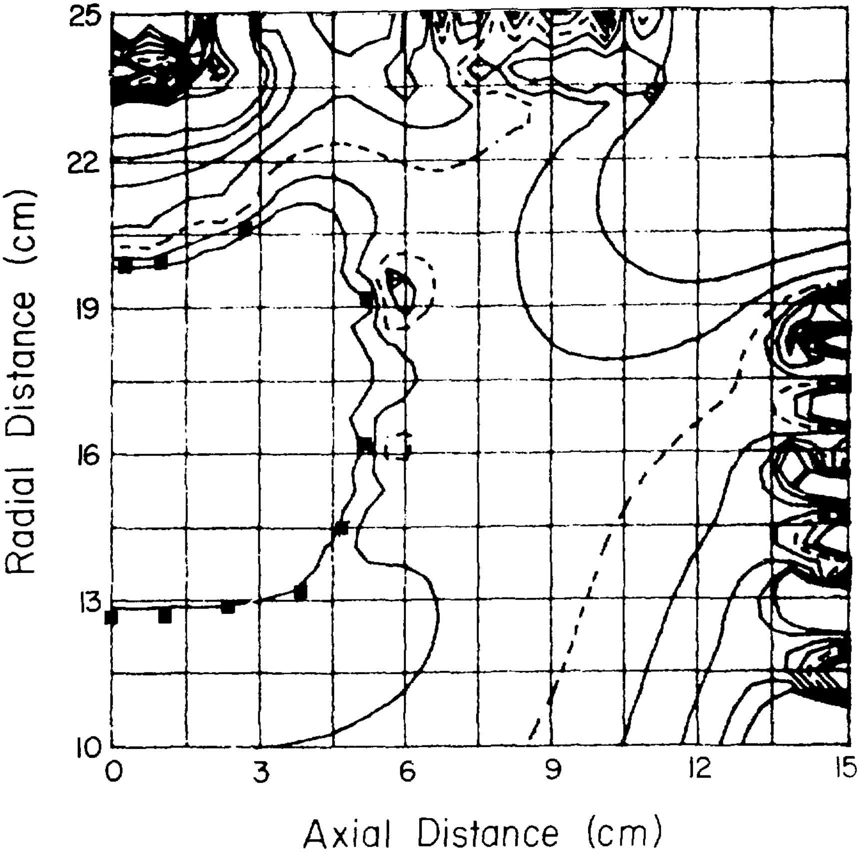

25
22
19
16
13
10
Radial Distance (cm)
0
3
6
9
12
15
Axial Distance (cm)
XBL 824-540

Figure 2b



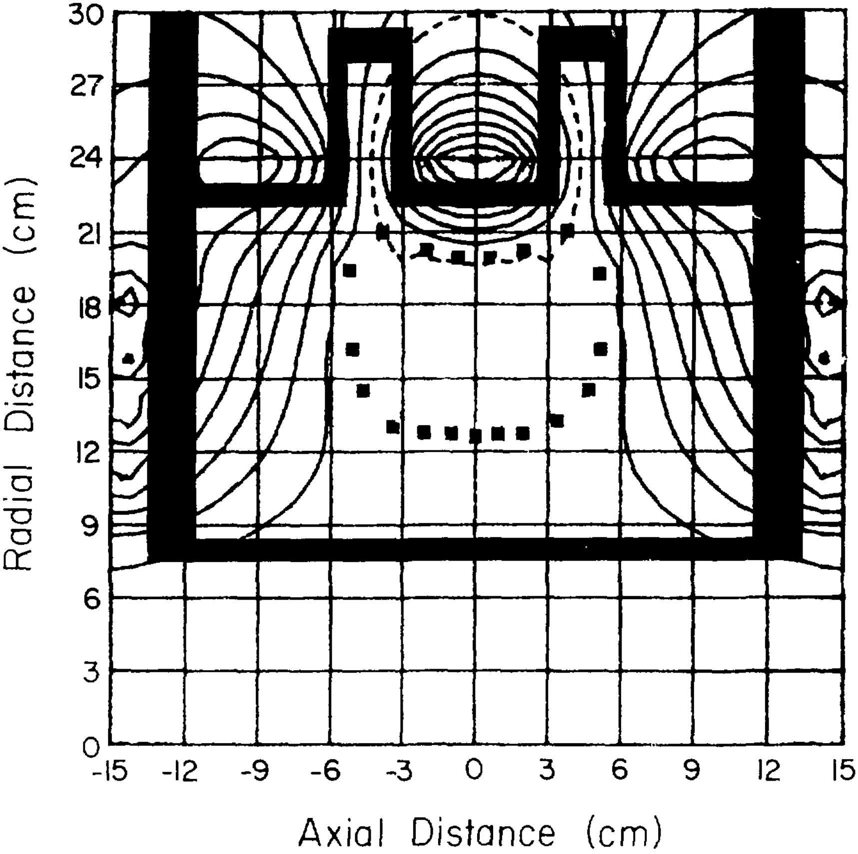
XBL 824 539

Figure 3a



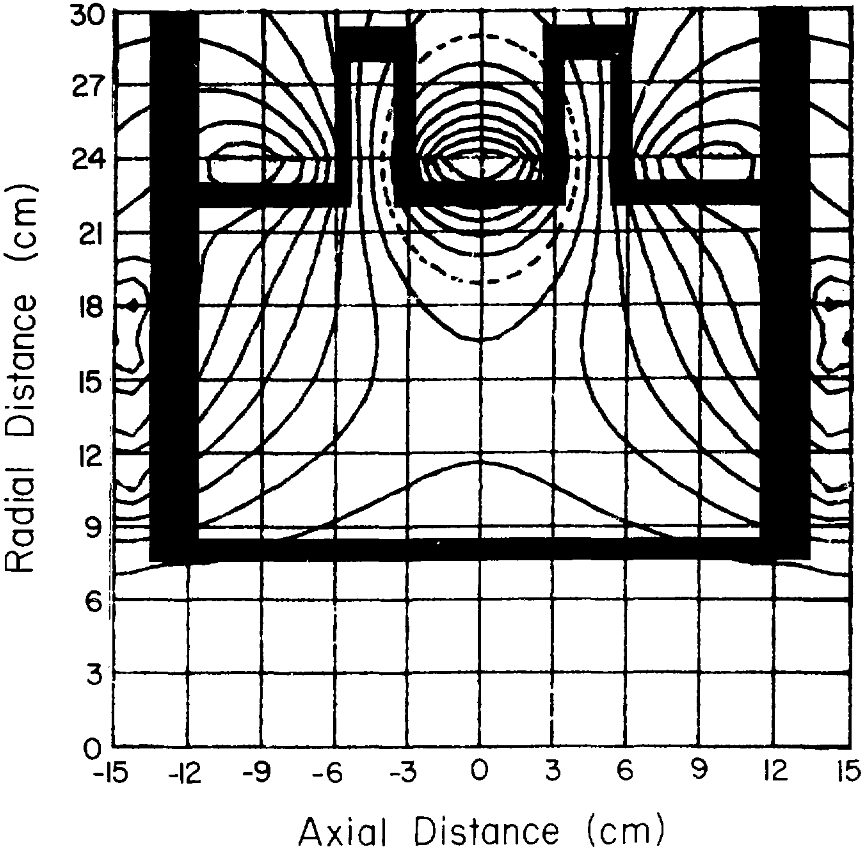
XBL 824-542

Figure 3b



XBL 824-543

Figure 4



XBL 824-538

Figure 5

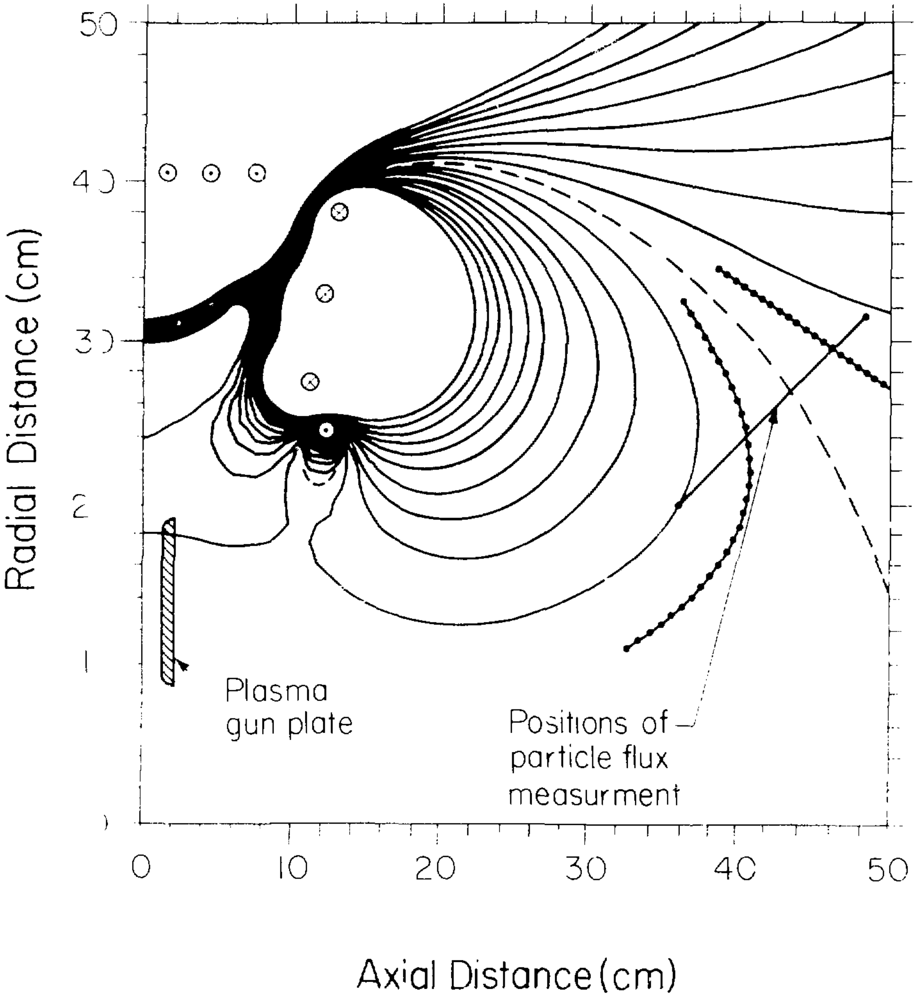

50
40
30
20
10
0
Radial Distance (cm)
0
10
20
30
40
50
Axial Distance (cm)
Plasma gun plate
Positions of particle flux measurment
XBL 8010-2219

Figure 6

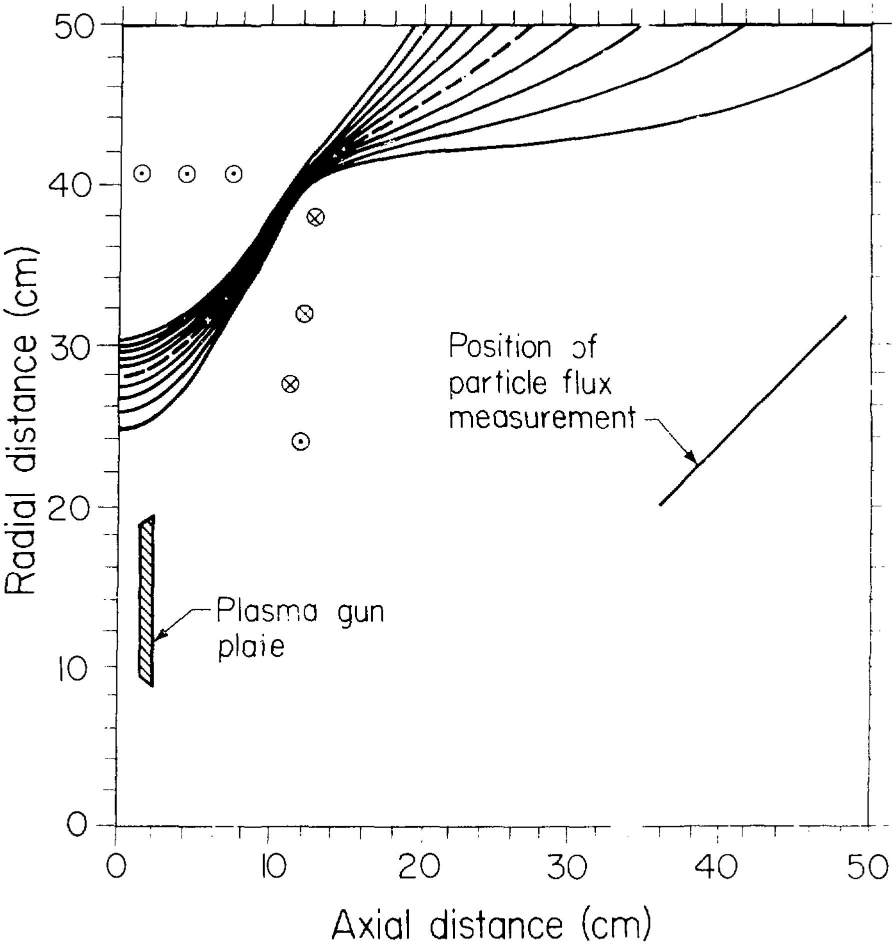

50
40
30
20
10
0
Radial distance (cm)
0
10
20
30
40
50
Axial distance (cm)
Position of particle flux measurement
Plasma gun plate
XBL 8010-2020

Figure 7