

# Lawrence Berkeley National Laboratory

## Lawrence Berkeley National Laboratory

### **Title**

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

### **Permalink**

<https://escholarship.org/uc/item/0w21z1fg>

### **Author**

Feinberg, B.

### **Publication Date**

1982-09-01

Peer reviewed

I-10927

LBL-10070  
UC-20g



# Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

## Accelerator & Fusion Research Division

BL--10070

LC83 016916

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

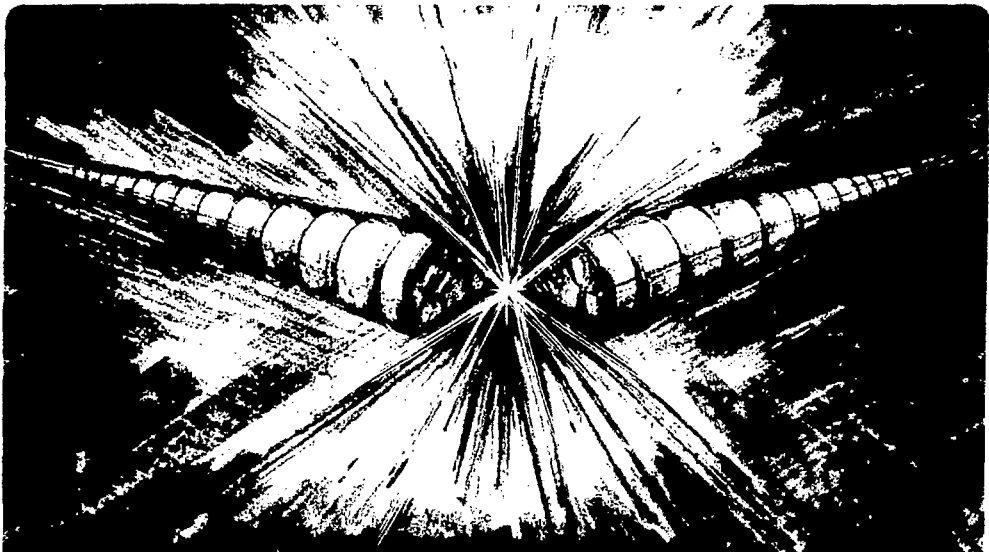
B. Feinberg, R.A. Niland, J. Coonrod,  
and M.A. Levine

September 1982

### NOTICE

PORTIONS OF THIS REPORT ARE ILLEGIBLE.

It has been reproduced from the best  
available copy to permit the broadest  
possible availability.



## A 2-D Skin Current Toroidal-MHD-Equilibrium Code\*

B. Feinberg, R. A. Niland,<sup>†</sup> J. Coonrod,<sup>‡</sup> and M. A. LevineLawrence Berkeley Laboratory  
University of California  
Berkeley, CA 94720

September, 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.

**DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

\*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.

<sup>†</sup>Present address: University of Sydney, Physics Dept., Sydney, Australia

<sup>‡</sup>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<sup>1</sup> was initially used to study toroidal magnetic cusp formation and equilibrium.<sup>2</sup> 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,  $\beta=1$  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<sup>3</sup> and Tormac P-1<sup>4</sup>, and data interpretation of Tormac P-1<sup>5</sup> 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,  $\Psi$ , 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<sup>6</sup> for force balance, along with the relevant steady state Maxwell's equations.

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

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

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

Equations (1) and (2) yield

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

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,

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

Equation (3) implies that we can define the poloidal flux function  $\psi = RA_\phi$ , where  $A_\phi$  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  $\psi$  value on the plasma boundary. More detailed instructions on setting the  $\psi$  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

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

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<sup>(7)</sup> are used to calculate  $\psi_i$  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<sup>(8)</sup>. Once all the mutual inductances are known, one can, therefore, write the matrix equation

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

The summation is over all currents  $I_j$  and inductances  $M_{ij}$ , and yields the poloidal flux function  $\psi$  at each position  $i$ . Since the value

of  $\psi$  at the plasma surface is input,  $\psi$  at the coil locations representing the plasma boundary is known. The poloidal flux function,  $\psi$ , 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  $\psi$  until one has the desired total current. Since the equations are linear, scaling a single trial  $\psi$  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.<sup>9</sup>

It should be noted that the inductance matrix is ill conditioned. This means that small changes in  $\psi_j$  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  $\psi$ . 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  $\psi$  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

$$P_0 = nkT = (B_T^2 + B_p^2 - B_{Ti}^2)/8\pi \quad (8)$$

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,  $R_0$ , the poloidal field,  $B_p$ , is negligible in a bicuspl such as Tormac. One can therefore write

$$8\pi P_0 = (B_{T_0}^2 - B_{Ti_0}^2)$$

Equation 8 can be written as

$$8\pi P_0 = (B_{T_0}^2 - B_{Ti_0}^2) \frac{R_0^2}{R^2} + B_p^2$$

Therefore

$$B_p^2 = 8\pi P_0 \left(1 - \frac{R_0^2}{R^2}\right)$$

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  $8\pi$ .

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  $\psi$ , or poloidal magnetic field lines, are available as well as contours of constant  $|B|$ . The user determines the range of  $\psi$  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  $\psi$  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  $\psi$  is calculated as described in Step C, through the use of a second order polynomial approximation for the elliptic integrals.<sup>7,8</sup>  $|B|$  is calculated from the  $\psi$  values using  $\vec{B} = \vec{\nabla} \times \vec{A}$ , 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.<sup>10</sup>

### 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<sup>3</sup> and Tormac P1<sup>4</sup>. Tormac P1 was built and operated, and the computer code played an integral role in the interpretation of the experimental observations.<sup>5</sup>

Tormac VB design presented some special problems. The goal was to build an experiment about the size of Tormac IV<sup>11</sup> (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.<sup>5</sup> 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  $\psi$  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.OBL TO LINK
C-----THIS PROGRAM WITH SUBPROGRAMS EPSICAL,BBFF,FCURR,
C-----BERRA AND EPLIN, AS WELL AS GRAF AND CONTURB.
C-----THIS PROGRAM HAS THE CAPABILITY OF MODIFYING 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/PL(50),ZL(50),IL,NE,IPLAME(8),PSIG
COMMON/BLK2/CR(50),CZ(50),CI(50),NC,IVAME(3)
COMMON/BLK3/ACR(50),ACZ(50),ACI(50),NAC,ICAME(8)
COMMON/BLK5/IPSAMT(8),CURZ,IFLG,JFLG
COMMON/BLK6/RSTEP,ZSTEP,IRSTRT,IZSTRT
COMMON/BLK7/P(50),RN(50),ZN(50),DLB(50),PI
DATA INX,ICO,IPL,ICP,MP,MI/2HEX,2HCC,2HPL,2HCP,2HNP,2HNI/
DATA NG,IFI,IBR/2HNO,2HPI,2HBR/
IGS=81
IBL=7
IPLFLG=2
C
C-----INPUT GRID INFORMATION
30 CALL TVOL
   TYPE 30,IGS
35 FORMAT(1X,A1,'INPUT START R, END R, START Z, END Z (CM,10:0,10:
   1-10,20:1)')
   READ(5,40,ERR=30)IRSTRT,IREAD,IZSTRT,IZEND
40 FORMAT(4I4)
   RSTEP=(IREAD-IRSTRT)/40.
   ZSTEP=(IZEND-IZSTRT)/40.
C-----INPUT CENTER CONDUCTOR
50 TYPE 50
55 FORMAT(' INPUT CENTER CONDUCTOR(?)')
   IFLG=0
   CALL COIL(1,ICAMP,ACR,ACZ,ACI,NAC)
   IF(NALC.EQ.2)IFLG=1
C
C-----INPUT EXTERNAL COILS
60 TYPE 60
65 FORMAT(' INPUT EXTERNAL COILS(?)')
   CALL COIL(2,IVAME,CR,CZ,CI,NC)
C
C-----INPUT Z CURRENT
70 TYPE 70
75 FORMAT(' INPUT Z CURRENT (AMPS) (10:3.34) ',5)
   READ(5,75,ERR=70)ICUPZ
78 FORMAT('11.0')

```



```

C
C-----INPUT THE PSI FILE
CJFLG=
C TYPE=22
CFCRMA1 ' INPUT PSI FILENAME OF "NO"?',($)
C READ(5,-4,ERR=300)IPSAME
CFCRMA2($2)
C IF(IPSAME/1)NE.NC)JFLG=1
C
C-----BRANCHING SECTION
C TYPE=17,IGS
CFCRMA1(1Y,1A1,'02 PLASMA,0P CHANGE PLASMA,NE=NEW
C 1 FIIP' NI NEA 7 CURRENT,0C-COILS ONLY,EX=EXIT ',($)

CFCRMA2(2,0C,ERR=BRANCH)
CFCRMA1(1A2)
C CALL TVOL
C IF(02)GOTO(1P)GOTO 20
C STOP '---EXIT'
CFCRMA2(1P)GOTO 157
CFCRMA2(1P)GOTO 200
CFCRMA2(1P)GOTO 32
CFCRMA2(1P)GOTO 77
CFCRMA2(1P)GOTO 250
C
C-----INPUT PLASMA & SET CURRENTS
CFCRMA1(1A2)
CJFLG=1
C TYPE=167,IGS
CFCRMA1(1X,1A1,'01 DISPLAY CONFIGURATION,RR=BRANCH ',($)
C READ(5,-4,ERR=100)INCMN
CFCRMA2(1P)GOTO 35
CFCRMA1(1A2)
C
C-----CHANGE PLASMA & SET CURRENTS & PBA
CFCRMA1(1Y,1A1,'01 TYPE 017,IGS
CFCRMA1(1X,1A1,'01 IN PLASMA BEFORE CHANGING PLASMA'
C IF(1P)GOTO 35
C CALL TVOL
CFCRMA1(1A2)
C
C-----COILS ONLY
CFCRMA1(1A2)
C TYPE=168,IGS
CFCRMA2(2,0C,ERR=BRANCH)
CFCRMA2(1P)GOTO 35
CFCRMA1(1A2)
CFCRMA1(1A2)
CFCRMA1(1A2)

```

```

0
0
0-----SUBROUTINE TO ENTER COILS
0
      SUBROUTINE COIL(M,NAMF,RL,ZL,IL,NL)
0--      M=1 > INPUT R,Z,I FILE
0--      M=1 > INPUT R,Z FILE
0--      M=2 > OUTPUT R,Z FILE
0--      M=3 > OUTPUT R,Z,I
0
      REAL IL(50),RL(50),ZL(50)
      INTEGER NAME(8),INAME(8)
      DATA NO-ZHNO/
      IGS=11
      IF(M.EQ.2)GOTO 300
      IF(M.EQ.3)GOTO 200
0
0-----INPUT DATA FILE
10      TYPE 15
15      FORMAT(' COIL DATA FILE?(NAME OR NO) ',9)
20      READ(5,20,ERR=10)INAME
25      FORMAT('A2')
0
0-----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,NAMF,16)
0
0-----BRANCH IF M=1 (JUST R & Z)
      IF (M.EQ.1)GOTO 40
40      READ(1,25,ERR=10)NL
45      FORMAT('I2')
48      READ(1,20,ERR=10)(RL(I),ZL(I),IL(I),I=1,NL)
50      FORMAT('F10.2')
55      CALL CLOSE(1)
      RETURN
0
0-----JUST R & Z
40      READ(1,25,ERR=10)NL
45      READ(1,20,ERR=10)(RL(I),ZL(I),I=1,NL)
48      FORMAT('F10.2')
50      CALL CLOSE(1)
      RETURN
0

```

```

C-----INPUT DATA AND BRANCH IF JUST R AND Z
00      IF(N.EQ.1)GOTO 250
01      TYPE 55
05      FORMAT(' ENTER NUMBER OF COILS (EG:19) ',S)
06      READ(5,25,ERR=50)NL
07      TYPE 57
07      FORMAT(' ENTER R(CM),Z(CM),I,ABAMP) (EG:5.21,10.7,50.5) ',S)
08      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" ? ',S)
211     READ(5,20,ERR=200)INAME
212     IF(INAME(1).EQ.N0) RETURN
213     DO 212 I=1,8
214     NAME(I)=INAME(I)
215     CALL ASSIGN(1,NAME,16)
216     *FITS(1,25)NL
217     *SIFR(1,30)(RL(I),ZL(I),IL(I),I=1,NL)
218     CALL CLOSE(1)
219     RETURN

C
C-----INPUT R AND Z DATA
250     TYPE 55
251     READ(5,25,ERR=250)NL
252     IF(NL.EQ.0)RETURN
253     TYPE 57
254     FORMAT(' ENTER R(CM),Z(CM) (EG:14.82,1.2) ',S)
255     READ(5,40,ERR=255)(RL(I),ZL(I),I=1,NL)

C
C-----OUTPUT R AND Z DATA FILE
300     TYPE 310
301     READ(5,20,ERR=300)INAME
302     IF(INAME(1).EQ.N0)RETURN
303     DO 310 I=1,8
304     NAME(I)=INAME(I)
305     CALL ASSIGN(1,NAME,16)
306     *FITS(1,25)NL
307     *SIFR(1,40)(RL(I),ZL(I),I=1,NL)
308     CALL CLOSE(1)
309     RETURN

```

C  
C

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

DATA A,B,C/1.3862944,.11198906,.772532303/  
DATA E,F,G/.5,.12134863,.028874723/  
Y=1.-X  
IF(Y.LE.1.E-4)Y=1.E-4  
P=A+Y\*(B+Y\*C)  
Q=E+Y\*(F+Y\*G)  
FLK=P-A\*LOG(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.,.46321053,.10773575/  
DATA E,F,G/.24527397,.241253211/  
Y=1.-X  
IF(Y.LE.1.E-4)Y=1.E-4  
P=A+Y\*(B+Y\*C)  
Q=Y\*(F+Y\*G)  
ELE=P-A\*LOG(Y)\*Q  
RETURN  
END

```

C      ***** FELIX.PRN *****
C
C-----SUBROUTINE TO CALCULATE CURRENTS TO MODEL PLASMA
C-----USER DETERMINES COIL LOCATIONS AND PFI DETERMINES CURRENTS
C
C      SUBROUTINE PFI(M1)
C--      N1=2 > INPDE PLASMA
C--      N1=1 > CHANGE PLASMA
C--      N1=2 > VACUUM (CENTER CONDUCTOR CURRENTS)
C--      IFLG=0 > CENTER CONDUCTOR PRESENT
C--      IFLG=1 > NO CENTER CONDUCTOR
C
C      F=AD IL(BM),P(BM),RN1(BM),ZN1(BM),LLC(FM)
C      I=TELEP,KIN(BM),KNCT(BM)
C      COMMON/BLK1/PL(BM),ZE(BM),IL,NL,IPLAME(2),PSI(2)
C      COMMON/BLK2/DB(BM),CZ(BC),CI(BC),NO,IVAMB(4)
C      COMMON/BLK3/ACR(BM),ACZ(BM),ACI(BM),MAC,IACCANE(4)
C      COMMON/BLK4/IFSAME(8),CURZ,IPLG,JPLG
C      COMMON/BLK5/RN1,RN1,DLE,P1
C      COMMON/BLK6/BB,BZ
C      I=PI,IQG,MEG/Z.141F9,P.2/
C      IRL=7
C
C-----CALCULATE CENTER CONDUCTOR CURRENTS (5 OUTER COILS) ONLY
C-----IF M1=2 LABEL IFLG.FLDBOICALL CURR ACR,ACZ,ACI,MAC,MAC,M.1
C-----IF M1=1,2 RETURN
C
C-----TO CHANGE PLASMA TO I=4
C-----IF IFLG=1,2 RETURN
C
C-----INPUT 2 STATE VALUE OF PSI
C-----I=PP 17
C-----FORMAT(5XINT12,F31.4I PLASMA SURFACE (40:1.25) )
C-----M1=1,2,3,4: I=110
C-----I=PP 11.00
C
C-----INPUT 2 PLASMA
C-----M1=1: I=1,3
C-----I=1: I.
C-----I=2: I.
C-----I=3: I.
C-----I=4: I.
C-----CALL COIL(1,IPLAME,-L,IL,IL,M)
C
C-----CALCULATE PLASMA AND CENTER CONDUCTOR CURRENTS
C-----PLASMA
C-----I=PP 17.0 I CALL QUERRI,ZI,IL,NL,M,PSI(2)
C-----I=PP 17.1 SCOT=42
C

```

```

0-- 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,PSIG)
    DO 33 I=1,NAC
        K=NL+I
        ACI(I)=IL(K)
        RL(K)=0.
        ZL(K)=0.
        IL(K)=0.
33   CONTINUE
    C

```

0-----CALCULATE PLASMA PRESSURES

```

42   DO 52 K=1,NL
        K1=K+1
        KC=K-1
        IF(K.LD.1) KC=NL
        IF(K.GD.NL) K1=1
        P(K)=0.
        ZM=ZL(K)
        RM=RL(K)
        IP=PI*IL(K)
        IL(K)=0.
        CALL RPSIG(ZM,RM,PL,ZL,IL,NL,K)
        IL(K)=IP*PI
        BP1=BR
        BZ1=BZ
        CALL RPSIG(ZM,RM,CR,CZ,CI,NC,K)
        BP1=BR+BP1
        BZ1=BZ+BZ1
        IF(IEG.LD.1) GOTO 45
        CALL RPSIG(ZM,RM,ACR,ACZ,ACI,NAC,K)
        BP1=BR+BP1
        BZ1=BR+BP1
44   RT=2.*CURZ/ZM
        P(K)=P(K)+(BP1**2+BZ1**2+BT**2)/(2*3.14159)
        DL(K)=SQRT((PL(K1)-RM)**2+(ZL(K1)-ZM)**2)
52   CONTINUE
        IF(IEG.LD.2) GOTO 140
        IF(IEG.ND.NEG) GOTO 220
    C

```

```

C-----PRINT OUT PLASMA MODEL AND PRESSURE
114     TYPE 112,IPL,IVAME,IPLAME,IOCCAMP,CURZ
115     +CHMAT(IX,IA1,'FILE','PA2/' PLASMA=' PA2/' COND=' PA2
1 / Z CURRNT=' E9.3)
      CC 1% I=1,(NL+1)/2
      TYPE 122,I,PL(I),ZL(I),IL(I),P(I),LLS(I)
120     CONTINUE
125     +CHMAT(IX,'J','I','R','F6.2','Z','F6.2','I','F9.2',
1 / 'P','F9.2','D','F9.2)
      INM1.FO.LABRTURN

```

```

C-----INPUT CHANGES IN PLASMA POSITION
130     TYPE 140,NL
135     +CHMAT(IX,CHANG=K04,CHANG=LINPEL # OF LOGS ('0','1','2',
1 / '3) CHANG=JAPOT 1 / 3)
      +CHMAT(IX,NL)
140     +CHMAT(IX,NL)
      IF(NL==1) GO TO 150
      IF(NL==1) GO TO 160
      IF(NL==1) GO TO 170
      IF(NL==1) GO TO 180
      IF(NL==1) GO TO 190
      IF(NL==1) GO TO 200
      IF(NL==1) GO TO 210
      IF(NL==1) GO TO 220
      IF(NL==1) GO TO 230
      IF(NL==1) GO TO 240
      IF(NL==1) GO TO 250
      IF(NL==1) GO TO 260
      IF(NL==1) GO TO 270
      IF(NL==1) GO TO 280
      IF(NL==1) GO TO 290
      IF(NL==1) GO TO 300

```

```

C-----
130     TYPE 140,NL
135     +CHMAT(IX,CHANG=K04,CHANG=LINPEL # OF LOGS ('0','1','2',
1 / '3) CHANG=JAPOT 1 / 3)
      +CHMAT(IX,NL)
140     +CHMAT(IX,NL)
      IF(NL==1) GO TO 150
      IF(NL==1) GO TO 160
      IF(NL==1) GO TO 170
      IF(NL==1) GO TO 180
      IF(NL==1) GO TO 190
      IF(NL==1) GO TO 200
      IF(NL==1) GO TO 210
      IF(NL==1) GO TO 220
      IF(NL==1) GO TO 230
      IF(NL==1) GO TO 240
      IF(NL==1) GO TO 250
      IF(NL==1) GO TO 260
      IF(NL==1) GO TO 270
      IF(NL==1) GO TO 280
      IF(NL==1) GO TO 290
      IF(NL==1) GO TO 300

```

```

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
227   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   PMOD1=SQRT(RN1(K)**2+ZN1(K)**2)
        RN1(K)=RN1(K)/PMOD1
        ZN1(K)=ZN1(K)/PMOD1
250   CONTINUE
C
C-----MOVE PLASMA
      DO 260 K=1,NL
        RL(K)=RL(K)+(P0-P(K))*PN1(K)*AISTEP*2.5E-9
        IF(K.EQ.1)GOTO 262
        ZL(K)=ZL(K)+(P0-P(K))*ZN1(K)*AISTEP*2.5E-9
262   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 28
284   CALL OCIL(0,IPLAME,RL,ZL,IL,NL)
      RETURN
      ENF
C
C

```



```

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

```

```

C-----CALCULATE BR,BZ AND PSI

```

```
BR=0.
```

```
ER=0.
```

```
Y(JJ)=0.
```

```
Z=Z1+.000001
```

```
R=R1+.000001
```

```
DO 100 L=1,NL
```

```
IF(IL(L))5,100,5
```

```
AZ=Z-ZL(L)
```

```
CAY2=(4.*RL(L)*R)/((RL(L)+R)**2+AZ**2)
```

```
PF=(.5*(2-CAY2)*EIK(CAY2)-FLF(CAY2))/SQRT(CAY2)
```

```
D=SQRT((RL(L)+R)**2+AZ**2)
```

```
D1=(RL(L)-R)**2+AZ**2
```

```
BF=BR+2.*IL(L)*AZ/(R*D)*(-FLK(CAY2)+(RL(L)**2+R**2+AZ**2)
```

```
1 *FLF(CAY2)/D1)
```

```
BZ=BZ+2.*IL(L)/D*(ELK(CAY2)+(RL(L)**2-R**2-AZ**2)
```

```
1 *FLF(CAY2)/D1)
```

```
Y(JJ)=Y(JJ)+IL(L)*PF*4.*SQRT(R*RL(L))
```

```
CONTINUE
```

```
Y(JJ)=-Y(JJ)
```

```
RETURN
```

```
END
```

```

C          ***** EPICAL.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 2.5 GRID INTERVALS
SUBROUTINE PSICAL(AR,AZ,BI,NA)
REAL AR(NA),AZ(NA),BI(NA)
COMMON/BLK6/RSTEP,ZSTEP,IRSTRT,IZSTRT
COMMON/BLK6/PSI(43,41),PE,PL
C
C-----CALCULATE GRID OF PSI VALUES
300 PL=1.E30
    PI=-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
                    C=(AR(L)+AJ)**2+(AI-AZ(L))**2
                    CAY2=C*4.
                    PF=(.5*(2.-CAY2)*FLX(CAY2)-ELF(CAY2))/SQRT(CAY2)
100     PSI(I1,J1)=PSI(I1,J1)+BI(L)*PF*4*SQRT(AJ*AR(L))
110     PH=AMAX1(PSI(I1,J1),PH)
        PL=AMIN1(PSI(I1,J1),PL)
RETURN
END

```

```

C      ***** FOURN.FTM *****
C
C-----SUBPROGRAM TO CALCULATE CURRENTS INDUCED IN A SET OF
C-----THEROIDALLY 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. KRESHAH, 'J. COMP. PHYS.', 26,43, (1978)
C-----SUBROUTINE CURR(UR,UZ,JI,NU,LU,U0)
C-----REAL UR(NU),UZ(NU),UI(NU)
C-----REAL*8 AM(30,30),AL(30,30),X(30),Y(30)
C-----COMMON/BLKA/AM,Y
C-----COMMON/BLKP/UR(50),UZ(50),UI(50),NU
C-----GET INDUCTANCES
C-----DO 10 I=1,NU
C-----DO 10 J=1,NU
C-----CALL LMAT(I,J,UR,UZ,NU)
10     CONTINUE
C
C-----GET PSI'S FOR PLASMA AND...
C-----DO 20 II=1,LU
C-----CALL BPSIC(UZ(II),UR(II),CR,CZ,CJ,NC,II)
C-----Y(II)=Y(II)+U*
20     CONTINUE
C-----IF(LU.EQ.NU)GO TO 25
C
C-----...FOR CENTRE CONDUCTOR
C-----DO 25 II=LU+1,NU
C-----CALL BPSIC(U/(II),UR(II),CR,CZ,CJ,NC,II)
25     CONTINUE
C
C-----NORMALIZE THE INDUCTANCE MATRIX AND PSI'S
C-----CHAMP=3.
C-----DO 30 I=1,NU
C-----DO 30 J=1,NU
C-----TEM=ABS(A*(I,J))
C-----IF(TEM.GT.CHAMP)CHAMP=TEM
30     CONTINUE
C
C-----DO 40 I=1,NU
C-----Y(I)=Y(I)/CHAMP
C-----DO 40 J=1,NU
C-----X(I,J)=Y(I,J)/CHAMP
40     CONTINUE
C

```

```

0-----GET AL
      DO 57 I=1,NU
      DO 58 J=I,NU
0-----  NCIE I, NOT 1
          AL(J,I)=AM(J,I)
          IF(I.EQ.1)GOTO 59
          SUM=0.
          DO 45 K=1,I-1
45      SUM=SUM+AL(J,K)*AL(I,K)/AL(K,K)
0-----  *DO NOT REFER TO D EXPLICITLY, SINCE D(I)=1/AL(I,I)
          AL(J,I)=AM(J,I)-SUM
54      CONTINUE
0-----
0-----START BACK SUBSTITUTION
      X(1)=Y(1)/AL(1,1)
      DO 70 MI=2,NU
          SUM=0.
          DO 64 NI=1,MI-1
64      SUM=SUM+X(NI)*AL(MI,NI)

          X(MI)=(Y(MI)-SUM)/AL(MI,MI)
70      CONTINUE
0-----CONTINUE BACK SUBSTITUTION
      DO 80 MI=1,NU
          X(MI)=Y(MI)*AL(MI,MI)
80      CONTINUE
0-----
      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(MI,NI)
          X(MI)=(X(MI)-SUM)/AL(MI,MI)
100     CONTINUE
0-----
      DO 110 I=1,NU
          XI(I)=X(I)
110     CONTINUE
      RETURN
      END

```

```

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/BLK0/AM,Y
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)*PI)/Q
PF=(.5**2.-CAY2)*ELK(CAY2)-FIE(CAY2)/SQRT(CAY2)
AM(I,J)=PF*4.*SQRT(PI*R(J))
RETURN
END

```

```

C          ***** BURAW.FTN *****
C
C-----SUBPROGRAM TO DISPLAY PSI AND MOD P
C-----THIS PROGRAM ALSO PLOTS E VS R, AND B VS Z
C-----THE GRID IN THIS VERSION IS 40 X 40
C-----THIS PROGRAM IS TO BE USED IN CONJUNCTION WITH
C-----THE MAIN PROGRAM EPLAPO, AND WITH EPSICAL, RBE-, EPLIN, EOURA
C-----AND IS TO BE LINKED WITH GRAF AND CONTURB
C-----THE PSI AND B VALUES ARE CALCULATED EVERY GRID POINT
C
C          SUBROUTINE DRAW(M3)
C--      M3=0 > PLASMA
C--      M3=1 > VACUUM
C
C          REAL IL(50),FNZ(43),F(43,41)
COMMON/BLK0/PSI(43,41),PH,PL
COMMON/BLK1/B(43,41),BH,BL
COMMON/BLK1/PL(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),AAC,ICAME(8)
COMMON/BLK5/IPSAME(8),CURZ,IFLG,JFLG
COMMON/BLK6/RSTEP,ZSTEP,IRSTRI,IZSTRI
DATA MP,MB,MX,MG,MZ,PI,NC/ZHP ,2HP ,ZFLX,2HBR,2HBZ,3.14159,2HNO/
IGS=31
IBL=?
C
C-----CALCULATE PSI AND B MATRICES
C          DO 305 I=1,43
C          DO 305 J=1,41
305     PSI(I,J)=0
C
C----- IF NO VACUUM PSI FILE GO TO 312
C          IF(JFLG.EQ.0)GOTO 312
C
C----- INPUT VACUUM PSI FILE
C          CALL ASSIGN(1,IPSAME,16)
C          READ(1,FPR=30)PSI,PL,PH
C          CALL CLOSP(1)
C          GOTO 320
306     RETURN
C
C----- CALCULATE VACUUM PSI, AND OUTPUT FILE
312     CALL PSICAL(CR,CZ,CI,NC)
314     TYPE 312,IGS,IBL
312     FORMAT(1Y,2A1,'OUTPUT PSI FILENAME=?OF ACQ ',9)
C          READ(5,314,FPR=311)IPSAME
314     FORMAT(=A2)
C          IF(IPSAME(1).EQ.NO)GOTO 320
C          CALL ASSIGN(1,IPSAME,16)
C          WRITE(1)PSI,PL,PH
C          CALL CLOSP(1)
C          JFLG=1

```

```

C
C----- IF NO CENTER CONDUCTOR GO TO 330
330  IF(1PLG.EQ.1)GOTO 330
      CALL PSICAL(ACP,ACZ,ACI,MAC)
C
C----- FOR VACUUM PLOT GO TO 350
350  IF(MZ.LT.1)GOTO 350
C
C----- CALCULATE PSI FROM PLASMA
      CALL PSICAL(RL,ZL,IL,AL)
      CALL BRK(0)
      GOTO 400
350  CALL BRK(1)
C
C-----CALCULATE THE TOTAL ENERGY OVER THE GRID
430  DO 110 J=1,41
110  ENZ(J)=0
      PEG=0
      DO 150 I=1,43
          DO 160 J=2,
160  ENZ(I)=ENZ(I)+R(I,J)**2**((J-1)*RSTEP+IRSTPT)*RSTEP
          ENZ(I)=ENZ(I)+R(I,1)**2*IRSTPT*RSTEP/2.
          ENZ(I)=ENZ(I)+R(I,41)**2*(40.*RSTEP+IRSTPT)*RSTEP/2.
          IF(I.EQ.3)GOTO 140
          IF(I.EQ.43)GOTO 140
          ENG=ENG+ENZ(I)*2*STEP
          GOTO 150
140  ENG=ENG+ENZ(I)*STEP/2.
150  CONTINUE
      PEG=PEG/5.47
C
C-----PROVIDE MISCELLANEOUS DISPLAYS
370  CALL TYPN(2,1,80)
380  TYPN(2,165,161)
390  FORMAT(1X,2A1,' DISPLAY OF PSI,B R,EX-EXIT,PR=R VS
400  1 2 3 4 5 6 7 8 9 0
410  1 2 3 4 5 6 7 8 9 0
420  1 2 3 4 5 6 7 8 9 0
430  CALL TPCB
440  CALL TPCB INK=BT,WA
450  CALL TYPN(3,1,80)
460  TYPN(3,165,161)
470  TYPN(3,165,161)
480  CALL TPCB
C
C-----PRINT FOR THE APPROPRIATE DISPLAY
490  IF(ENP.EQ.1) GO TO 20
500  IF(ENP.EQ.2) GO TO 30

```

```

C--- PSI DISPLAY
30 TYPE 32,PL,PH,BL,BH,NCMD
32 FORMAT(' PSILO,PSIHI'/2E11.3/' BLO,BHI'/2E11.3
1 /' BMIFR ',1A2,' N.LO,HI'/' (EG:20,-1.E6,1.E6)'
2 /' N=EV*N # OF CONTOURS'/' RETWD=N LG & HI')
READ(5,11,FER=32)NA,AL,AH
AN=NA
11 FORMAT(I3,2E11.3)
ADOT=(AH+AL)/2
AGAP=(AH-AL)/AN
TYPE 12,ADOT,AGAP,IVAME,IPLAME,IUAME,CURZ,ENG,PSI
12 FORMAT(' DOT=',E11.3/' SPACF=',E11.3/' FILE=',8A2/' PLASMA=',8A2
1 /' COND=',2A2/' 2 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(1320,I)
J=10*I-40/174*IRSTRT+IRSTRT
TYPE 55,IGS,J
55 FORMAT(1X,1A1,I2)
50 CONTINUE

DO 59 I=275,995,174
CALL TVPU(I,25)
J=12*(I-275)/174*IZSTRT+IZSTRT
TYPE 57,IGS,J

57 FORMAT(1X,1A1,I3)
59 CONTINUE
C-----DRAW "X'S" REPRESENTING PLASMA BORDER
IF(M3.EQ.1)GOTO 66
DO 65 I=1,N1
BZ=(ZL(I)-IZSTRT)/ZSTPD+3
BR=(RL(I)-IRSTRT)/RSTPD+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,N0
BZ=(ZL(I)-IZSTRT)/ZSTPD+3
BR=(RL(I)-IRSTRT)/RSTPD+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(NCMD.EQ.ME)CALL CONTR(PSI,43,41,AL,AH,N)
IF(NCMD.EQ.ME)CALL CONTR(B,43,41,AL,AH,N)
GOTO 217

```



```

3
0-----GRAPH OF R VS R
72  TYPE 22,BL,BF
73  FORMAT(' ',B10,BH1//2E11.3// ' ENTER BH1,Z// ' (EG:3.E4,Z.) ')
    RPAT(5,13,ERR=70)AH,Z
13  FORMAT('11.3,F6.2)
    TYPE 74,CURZ,IVAME,IPLAME,PS10,Z
74  FORMAT(' Z CURRENT=' ,E9.3// ' FILE=' ,PA2// ' PLASMA=' ,PA2/
    1// PS10=' ,E9.3// ' B VS R// ' Z=' ,F6.2)
    START=IRSTRT
    ASTOP=IRSTPT+40.*RSTEP
    CALL GRID(START,ASTOP,300,1000,20,0.,AH,50,750,20)
    DO 90 I=200,995,174
    CALL TVPU(I,25)
    J=10*(I-200)/174*IRSTEP-IRSTPT
90  TYPE 55,IGS,J
    C
    DO 95 I=40,740,174
    CALL TVPU(100,I)
    J=(I-40)/174
    AJ=(AH)/4.*J
    TYPE 56,IGS,AJ
56  FORMAT(1Y,1A1,F11.3)
57  CCONTINUE
    IZ=(I-IRSTPT)/ZSTEP+Z
    CALL PLOT(START,B(IZ,1),1,0)
    DO 78 IR=1,41
    R=(IR-1)*RSTEP+IRSTPT
58  CALL PLOT(R,B(IZ,IR),1,1)
    CTO 200
    C

```

```

C-----GRAPH OF R VS Z
10      TYPE R2,R4,RH
92      FORMAT('  RIO,RHI'/2X11.3/' FENTER BHI,R'' (EG:3.E4,10.) ')
94      READ(5,17,FRR=90)AH,R
96      TYPE R4,CURZ,IVAMP,IPLAMB,PSI0
100     FORMAT('  Z CURPRINT=',F9.3/' FIIF=',E6Z///' PLASMA=',F4Z
102     1/' PSI0=',F9.3)
104     START=IZSTRT
106     NSTOP=IZSTRT+40.*ZSTEP
108     TYPE 502,R

500     FORMAT(1X///' B VS Z'///' R=',F6.2)
102     CALL SPID(START,NSTOP,200,1000,20,0.,AH,50,750,20)
104     DO 96 I=275,995,174
106         CALL TVPU(I,25)
108         R=14*(I-275)/174*ZSTEP+IZSTRT
110         TYPE 57,IG3,J

112         DO 98 I=40,740,174
114             CALL TVPU(150,I)
116             J=(I-40)/174
118             AJ=AH/4.*J
120             IF(XPLG.FO.1)AJ=(AH-AL)/4.*J+AL
122             TYPE 56,IG5,AJ

68      CONTINUE
124     IR=(R-IZSTRT)/ZSTEP+1
126     CALL PLOT(2.,B(3,IR),1,0)
128     DO 99 I7=3,47
130         Z=(I7-3.)*ZSTEP+IZSTRT
132         CALL PLOT(2.3(IZ,IR),1,1)
134     GO TO 277
136     END

```

```

C      ***** GRAPHIC *****
C--- THIS IS A SET OF ROUTINES FOR MAPPING AND PLOTTING
C--- DATA ON THE TEKTRONIX 4000 WHICH CAN BE CALLED
C--- FROM FORTRAN PROGRAMS.
C      JOHN COONEED      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,YLOG,YLOG
      LOGICAL/LS1/ANS(2)
      COMMON /SP1977/ XLOG,YLOG,XLO,YLO,DX,DY, SX,SY,IXL,IYL
      DATA XLOG,YLOG,IXL,IYL,SX,SY /FALSP,..,FALSP,..,FALSP,..,FALSP,..,FALSP,..,FALSP,..,FALSP,..,FALSP,..,
      ANS(1) /2007
      ANS(2) ,M
      IK=1
      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(N.LT.2) GO TO 10
      CALL WPU IX=I, IY=J
      CALL DO I0(ANS,2)
      GO TO 1
10    LF=OR(LF,LS1)OR(.NOT.GT.1)AND(.NOT.LT.10)
      IF(LF) CALL WPU(IX,IY)
      IF(.NOT.LF) CALL WVD(IX,IY)
      CONTINUE
      RETURN
      END
      SUBROUTINE GRID(X1,X2,IX1,IX2,N,Y1,Y2,IY1,IY2,M)
C--- THIS SUBROUTINE DETERMINES A LINEAR OR LOG MAPPING
C--- AND DETERMINES IT IN A CHIP.
C--- X1 --- X2 --- LINEAR GRID MARKS POINT X1 AND X2
C--- Y1 --- Y2 --- LOG MARKS PER DEC/DEC, (1500 DESIRE)
C--- FOR MANY DECADIS
C--- X1 --- X2 MARKS, LINEAR SCALE
C--- SIMILARLY WITH X FOR THE Y AXIS. THE LIMITS
C--- DEFINE THE CORNERS OF THE GRID ON THE 128X600 SCREEN.
      LOGICAL XLOG,YLOG,X1LOG,X2LOG
      COMMON /SP1977/ XLOG,YLOG,XLO,YLO,DX,DY, SX,SY,IXL,IYL
      CALL WPS(1977,..,FALSP,..)
      IXL=IX1
      IYL=IY1
      XLO=X1
      YLO=Y1
      CALL WPU N,LF,2
      XLOG=M,LF,2
      YLOG=N,LF,2
      X1LOG=X,LF,2

```

```

SX=(IX2-IX1)/(X2-X1)
IF(X1LOG) GO TO 30
IF(XLOG) GO TO 20
DO 10 J=1,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*12./NI)
DO 22 J=1,NJ
IY=IX1+DY*(J-1+AX)
CALL TVPU(IX,IY1)
CALL TVPD(IX,IY2)
22 SY=(IY1-IY2)/(Y1-Y2)
30 IF(Y1LOG) GO TO 100
IF(YLOG) GO TO 40
DO 32 I=1,N
IY=IY1+I*(IY2-IY1)/N
CALL TVPU(IX1,IY)
CALL TVPD(IY2,IY)
32 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(IY2,IY1)
DO 40 I=1,NI
AY=ALOG10(I*12./NI)
DO 40 J=1,NJ
IY=IY1+DY*(J-1+AY)
CALL TVPU(IX1,IY)
CALL TVPD(IX2,IY)
40 RETURN
END

```

```

SUBROUTINE TVCL
  LOGICAL*1 A(2)
  A(1)="253"
  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 GENADF(IPAR(1),A)
  IPAR(2)=N
  CALL CIO("410,5,1,,ISB,IPAR)
  CALL WAITFR(1)
  RETURN
END
SUBROUTINE TVPU(IX,IY)
  CALL TVCQ(IX,IY,1)
  RETURN
END
SUBROUTINE TVPD(IX,IY)
  CALL TVCQ(IX,IY,2)
  RETURN
END
SUBROUTINE TVCQ(IX,IY,I1)
  LOGICAL*1 A(2)
  I2=6-I1
  IF(I1.EQ.1)IY=0
  IF(IX.GT.1023)IX=1023
  IF(IY.EQ.0)IY=0
  IF(IY.GT.1023)IY=1023

```

```

  A(1)="253"
  A(2)="243.03.(IY/32)
  A(3)="343.03.(IY.AND.31)
  A(4)="143.03.(IX/32)
  A(5)="343.03.(IX.AND.31)
  CALL DOQIO(A(I1),I2)
  RETURN
END

```

```

C          ***** CONTOUR.FTN *****
C
C---CONTUR PLOTS NO UNIFARLY SPACED CONTOURS FOR APRAY
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 IF
C---CROSSES, AND JOINS THOSE POINTS WITH STRAIGHT LINES.
C---

```

JOHN JOONROD      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

```

```

A3="037"

```

```

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)

```

```

  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.NC) IC3=1

```

```

  IF ((IC1-IC3).GT.NC) GOTO 999

```

1

```

      DC 3 K=103,IC1
      LDOT=(K.EQ.IMIDC)
      EC 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)=(B(LL).GE.0.).AND.(Z(LL).LE.1.)
      S=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 TREC(Z,1000)
      TYPE 1999,IGS,IC3,IC1
1999  FORMAT(1X,1A1,2I6)
      RETURN
      ENL

```

3---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
      ENL

```

```

C      ***** 88E-10 *****
C
C----- SURPROGRAM TO CALCULATE B AT GRID POINTS
SUBROUTINE BDC(MVAC)
      REAL IL(50),BL(50)
      COMMON/BLK0 PSI(43,41),PH,PL
      COMMON/BLK1 B(43,41),BF,BL
      COMMON/BLK1/PL(50),ZI(50),IL,NL,IPLAME(6),PSIG
      COMMON/BLK1/IPSAME(6),CURZ,IPLG,JFLG
      COMMON/BLK6/RSTEP,ZSTEP,IRSTRT,IZSTRT
      COMMON/BLK7/P(50),RN(50),ZN(50),DLE(50),F1
C-----CALCULATE THE B ARRAY
      BL=1./F1
      RE=-BL
      DO 100 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)-6*(PSI(I-1,J1)-PSI(I-1,J1))/
1      (12*RSTEP*AJ)
      BZ=9*(PSI(I,J1+1)-PSI(I,J1-1))-(PSI(I,J1+2)-PSI(I,J1-2))/
1      (12*RSTEP*AJ)
      B(I,J1)=3*RT*BT+BF*BR+BZ*BZ)
      BU=AMAX1(BR,B(I,J1))
      BL=AMIN1(BL,B(I,J1))
125      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
720      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  $\psi$  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  $\psi$  at the plasma surface is input. The  $\psi$  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.

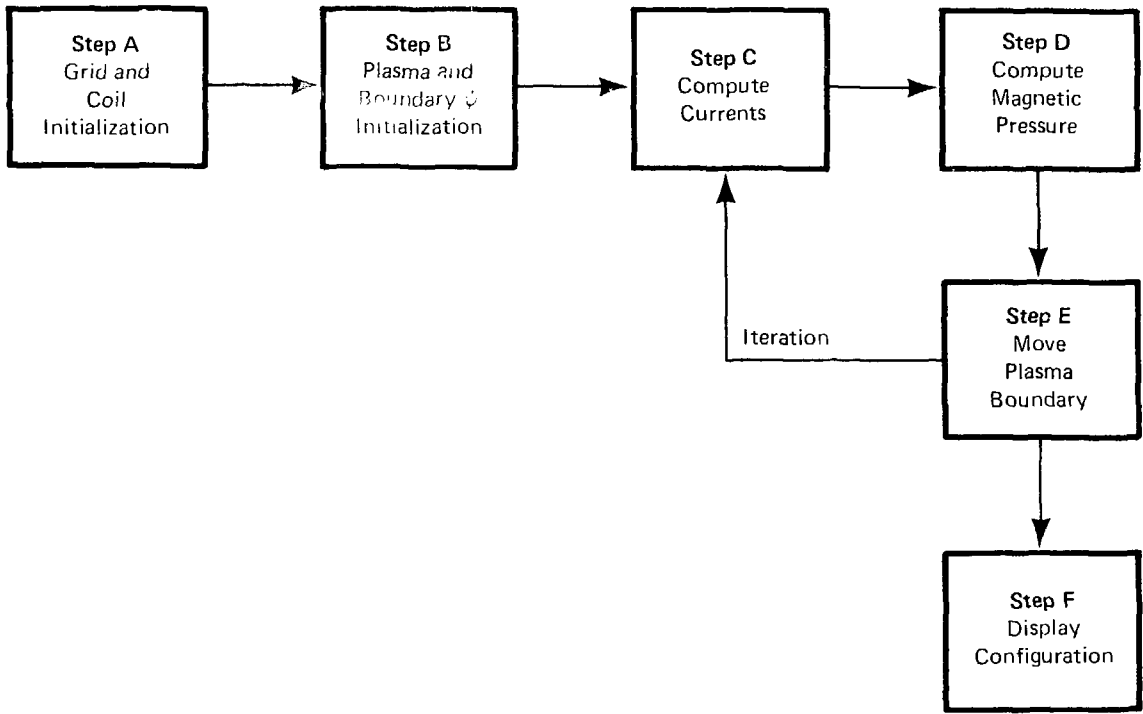
Other options in the branching section include "New File" which starts the program at the beginning, "New Z Current" which returns the program back to the toroidal field input, and "Coils Only" which

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 ( $\psi$ ) or  $\text{mod } |B|$  ( $|B|$ ) 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.

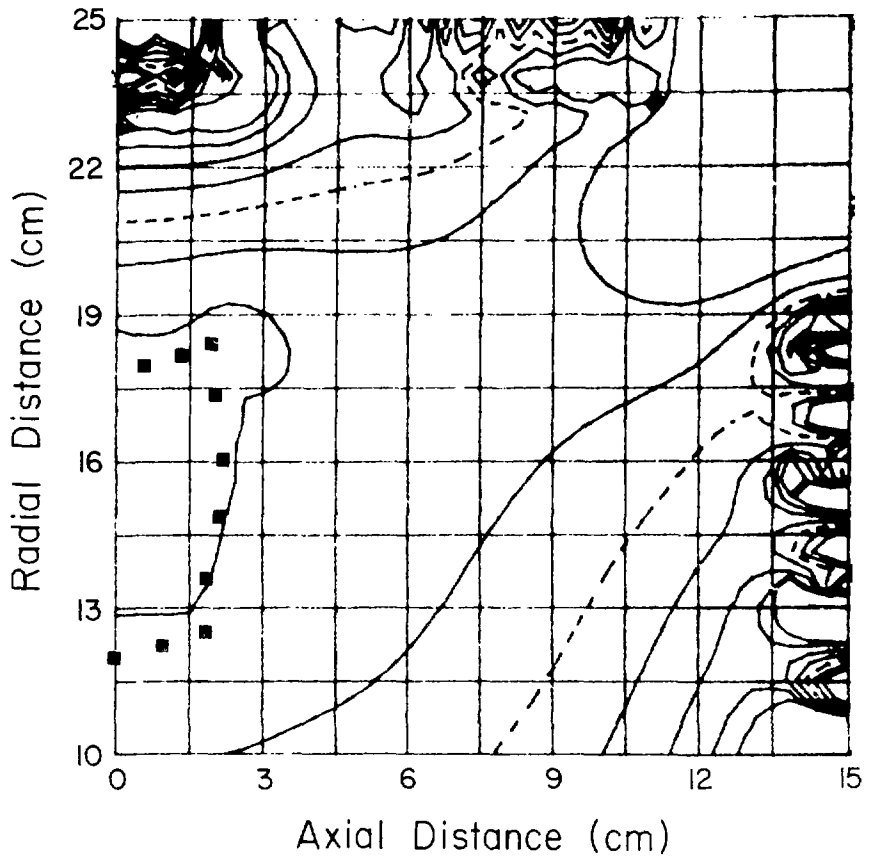
## Figure Captions

- 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.



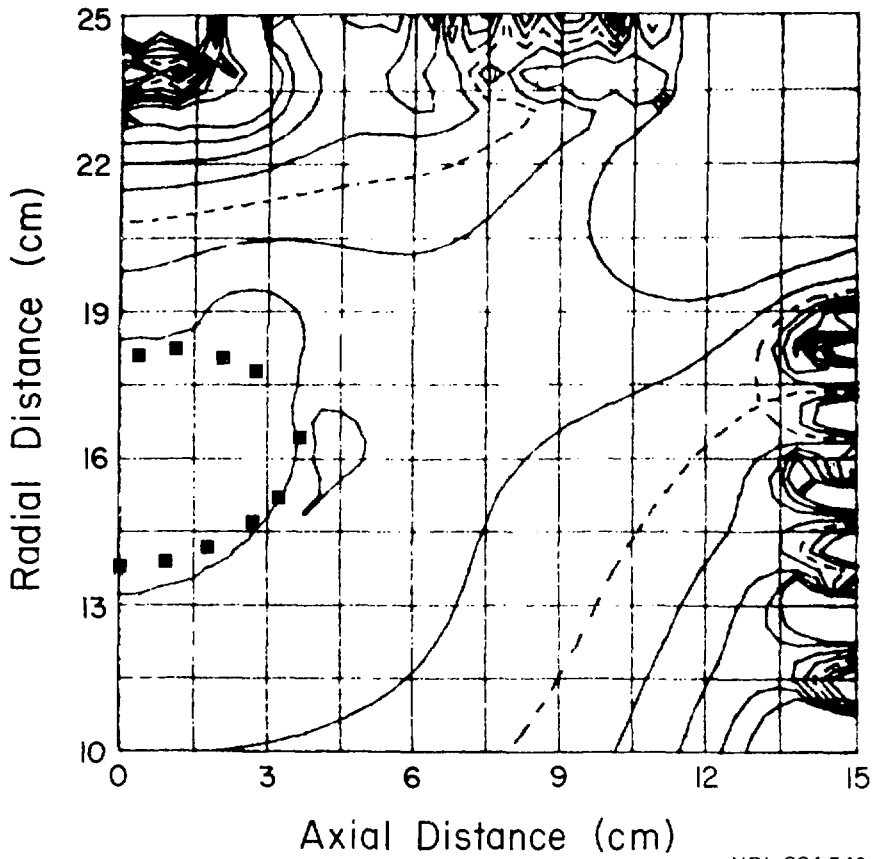
XBL 825-555

Figure 1



XBL 824 541

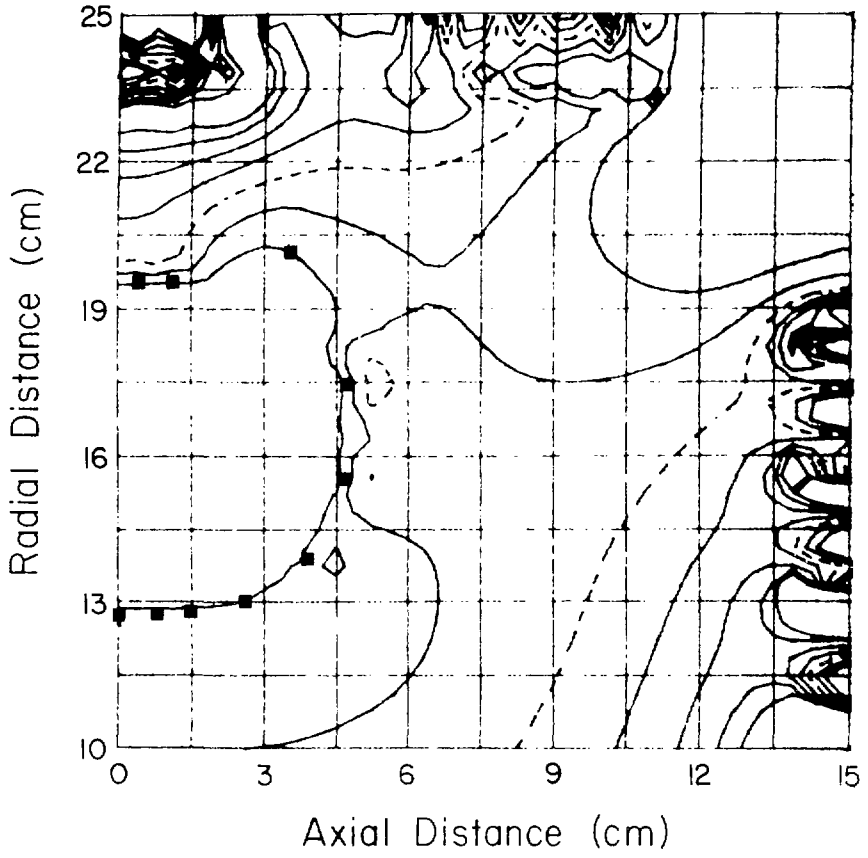
Figure 2a



XBL 824-540

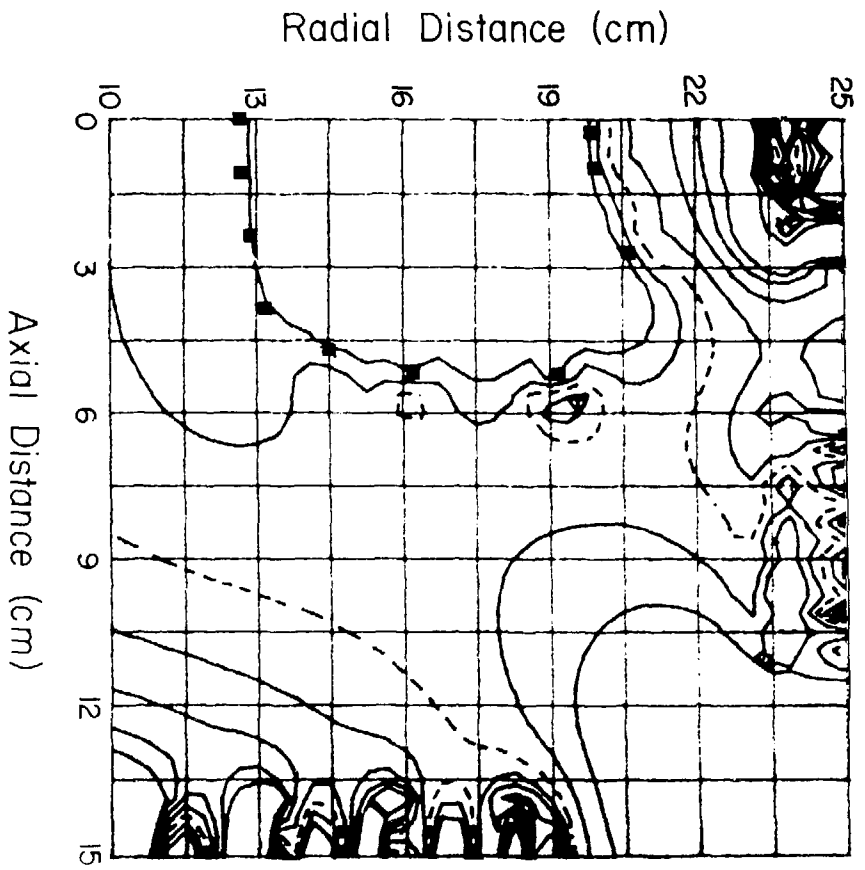
Figure 2b





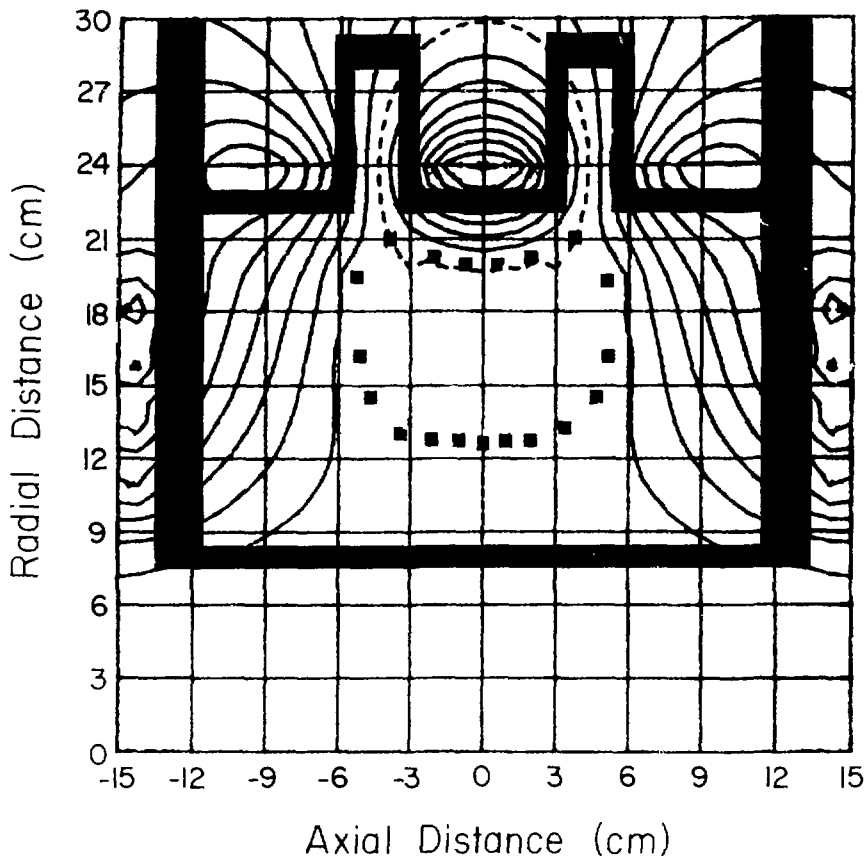
XBL 824 539

Figure 3a



XBL 824-542

Figure 3b



XBL 824-543

Figure 4

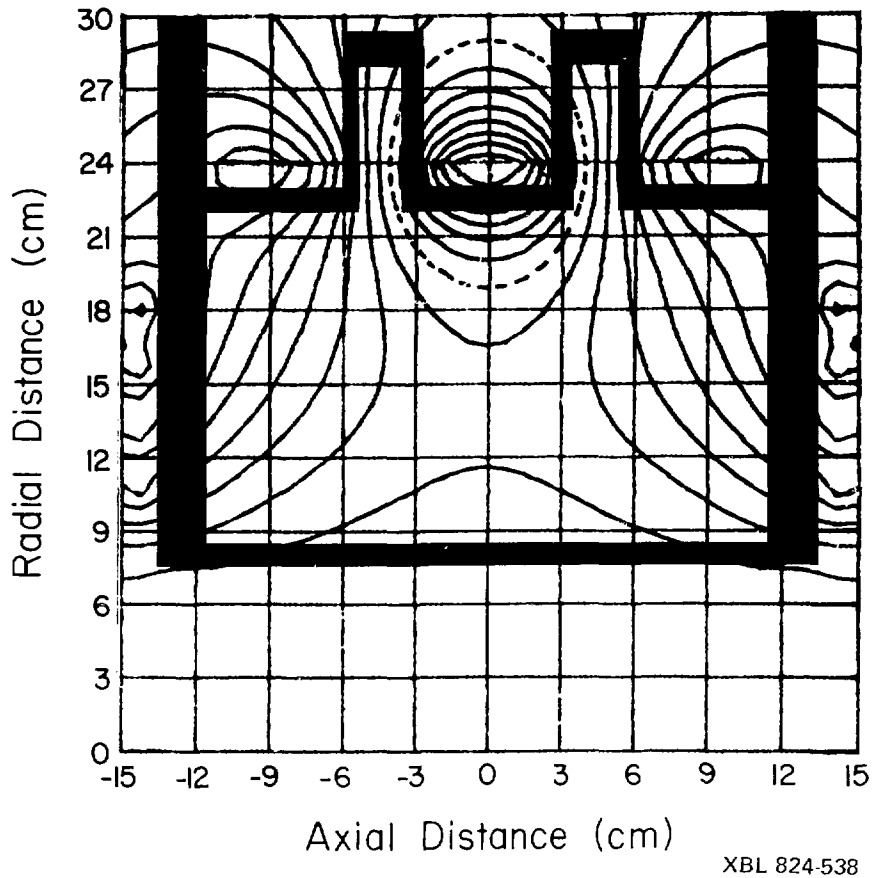
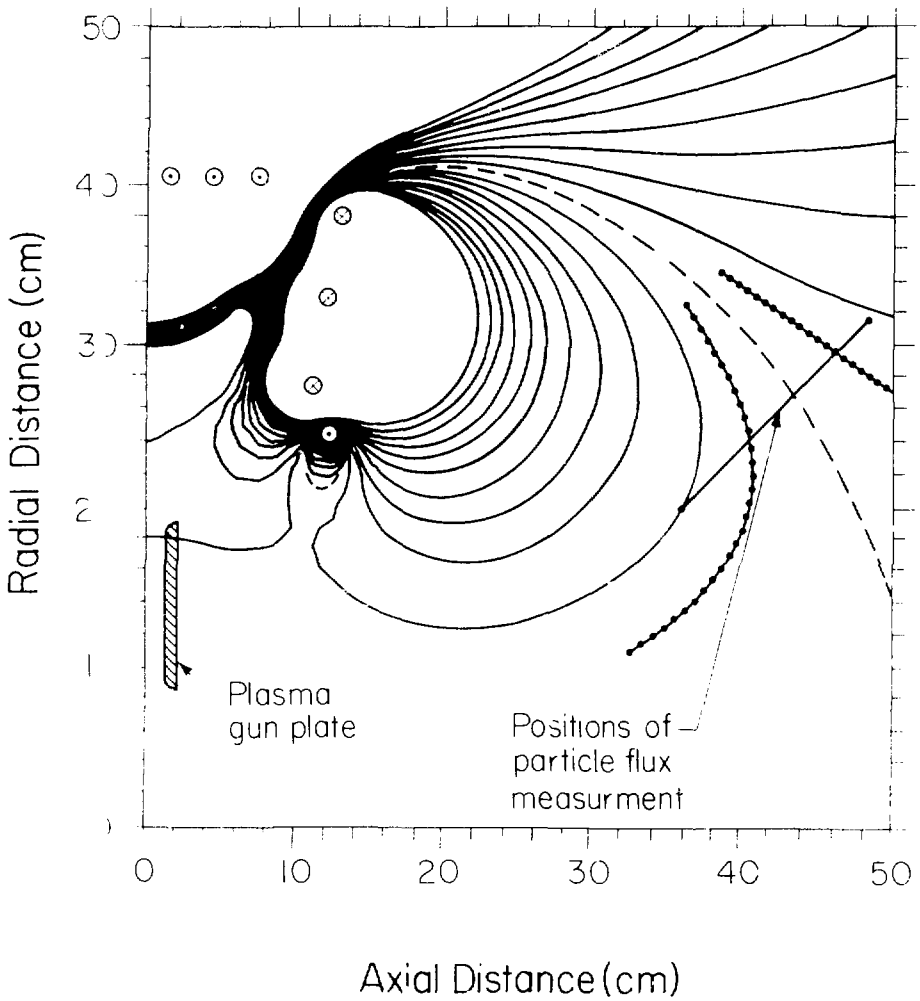
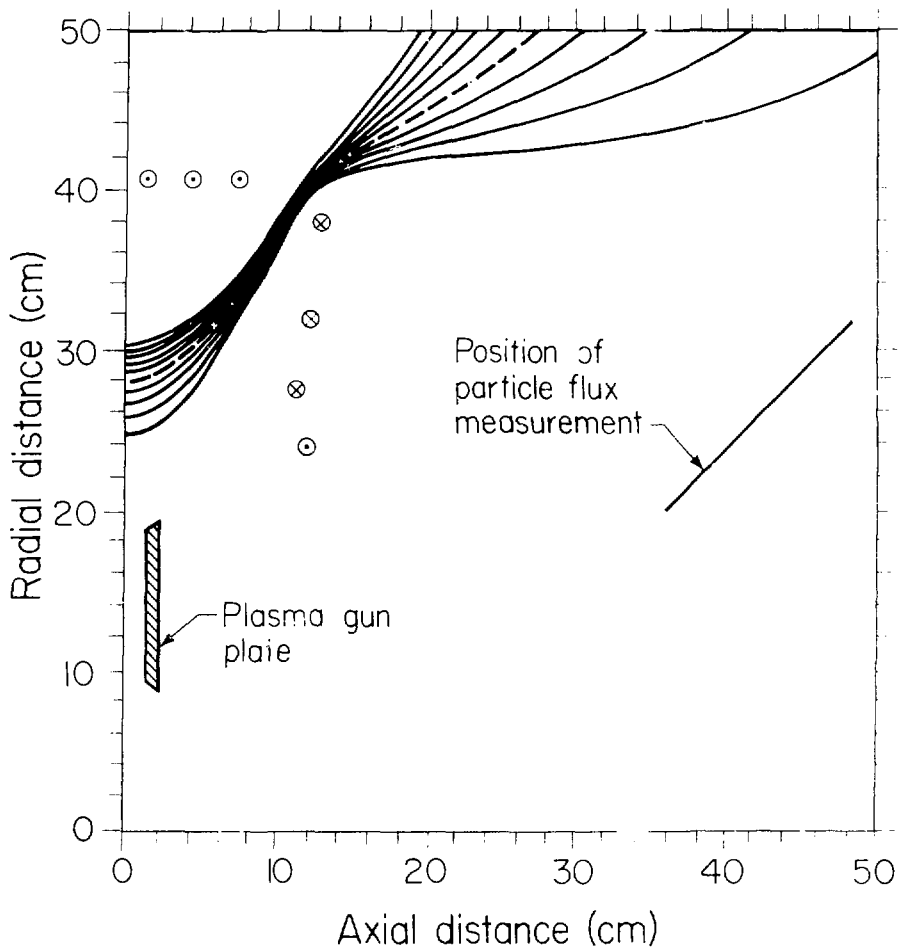


Figure 5



XBL 8010-2219

Figure 6



XBL 8010-2020

Figure 7

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.