189

3/13/81 T.S.

Dr. 2403

MASTER

ORNL/TM-7555

R-2710

ornl

OAK
RIDGE
NATIONAL
LABORATORY

UNION
CARBIDE

FLP: A Field Line Plotting Code for Bundle Divertor Design

Christoph Ruchti

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ORNL/TM-7555
Dist. Category UC-20 a, g

Contract No. W-7405-eng-26

FUSION ENERGY DIVISION

FLP: A FIELD LINE PLOTTING CODE FOR BUNDLE DIVERTOR DESIGN

Christoph Ruchti

Date Published - January 1981

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

CONTENTS

PREFACE

This work is the result of summer employment at Oak Ridge National Laboratory and was performed under the guidance of Dr. J. A. Rome.

ABSTRACT

A computer code was developed to aid in the design of bundle divertors. The code can handle discrete toroidal field coils and various divertor coil configurations. All coils must be composed of straight line segments. The code runs on the PDP-10 and displays plots of the configuration, field lines, and field ripple. It automatically chooses the coil currents to connect the separatrix produced by the divertor to the outer edge of the plasma and calculates the required coil cross sections. Several divertor designs are illustrated to show how the code works.

1. PURPOSE OF THE CODE

The code FLP is a tool used to design divertors. A listing of the code is contained in the attached microfiche. The main output consists of a plot showing the configuration of coils and magnetic field lines. As an additional feature, confined field lines can be followed halfway around the torus, and a ripple plot can be made. It is also possible to get a plot of B_y versus x along the x axis (the divertor is placed symmetrically with respect to the x axis).

2. HOW TO RUN THE CODE

The code needs two sources of input, each of which has a separate subroutine. In the subroutine INPUT, the configuration of coils is specified, and the code is told exactly what has to be done. The subroutine EQUIN reads the other source of input. It reads the equilibrium data and stores them in the common block /EQUI/. The data are computed by the Oak Ridge equilibrium code and are read from the file FOR25.DAT.

There are two modes of operation for the subroutine INPUT depending on the value of the variable ILGINP. The value is set by a DATA statement in the main program. For ILGINP = 1 the routine is in the long input mode and gets the data from the terminal assigned to channel 6. For each variable there appears a specific question on the screen. The toroidal field coils are specified implicitly, assuming the values for ETF. When all the data have been collected, they are written out onto file FOR16.DAT. The format is such that the file can later be read using the NAMELIST option. Every possible input variable is listed in this file, even if it is meaningless for the particular run. In this case, a default value is assigned.

For subsequent runs of the code it is more convenient to work in the short input mode with ILGINP = 0. In this case, the code reads the file FOR16.DAT. The values of the variables can easily be changed from run to run by editing this file.

The code can compute any number of configurations in one run. For each configuration there must be a $INDATA block on file FOR16.DAT, the first of which should be complete with all the variables listed. The following $INDATA blocks overwrite the preceding set of values, so that only altered values have to be listed.

3. MEANING OF VARIABLES

The important variables are grouped in common blocks. These blocks are discussed in the following:

/TFC/ This block contains the data concerning toroidal field coils. Each real variable has the prefix TF, and an integer variable starts with NTF. The variables TFRMID, TFWITH, TFHEIT, and TFCHOP specify the shape of the central filament of one coil as illustrated in Fig. 1.

TFTHIK and TFBROD give the rectangular cross section of a coil. There are NTF coils around the torus. NTFTIK and NTFBRD indicate how many filaments of current are used to calculate the magnetic field of one coil. A maximum number of NTFTIK = 5 filaments in toroidal direction and NTFBRD = 3 filaments in radial direction can be specified. These two numbers have to be odd. The variable TFCURF specifies the total current in all the toroidal field coils together; TFCURT is the current in one filament only. Finally, the arrays X0, Y0, and Z0 contain the corners of one toroidal field coil anchored in the (x,z) plane.

/TFCALT/ The code allows for the two toroidal field coils closest to the divertor to be changed. They can even be split up into two separate coils each. The specification is done by the same scheme as in /TFC/. Instead of the prefix TF, the altered variables have the prefix A. The shape of the altered coils differs from that of the standard toroidal

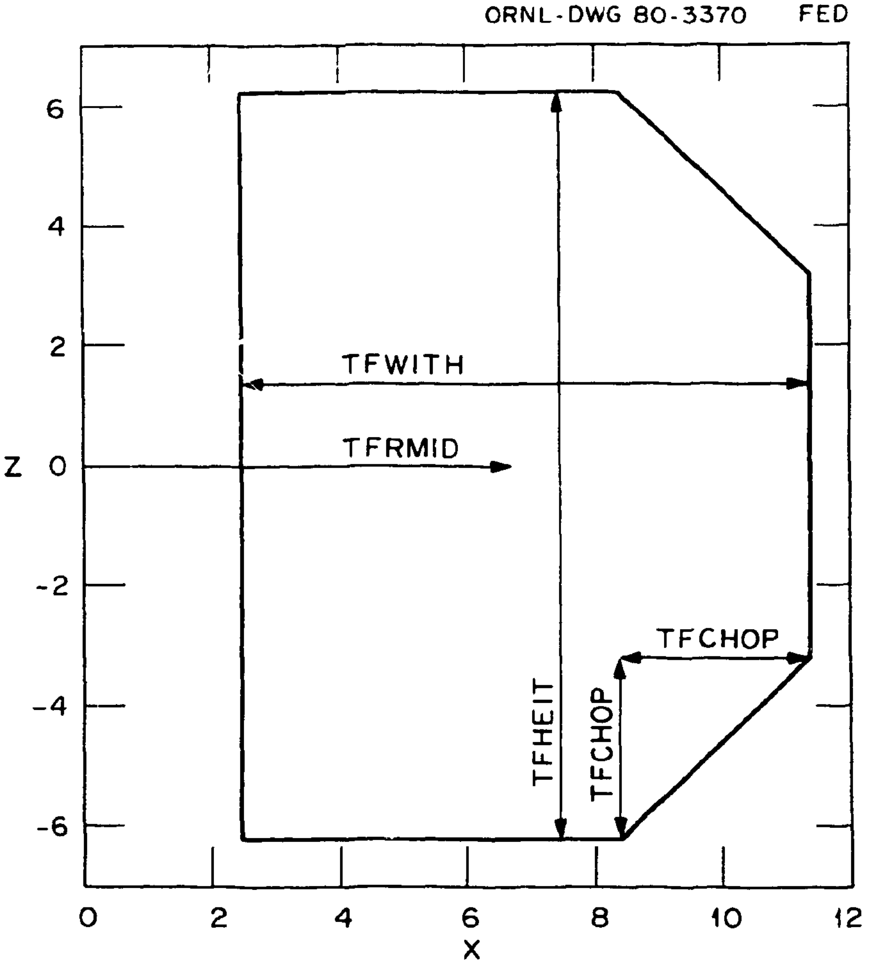


Fig. 1. Variables describing the central filament of a toroidal field coil.

field coils only in width. the variables TFHFIT and TFCHOP have no counterpart in the common block /TFCALT/.
The location of these altered coils is specified by the angles AFI1 and AFI2. They are measured in radians. For input/output, degrees are preferred, and the variables AFIDG, AFI2DG are used.

/DVC1/ This block contains the data about a one-coil divertor. All the variables have the prefix D1 or ND1, depending on their type. Divertor coils are rectangular, and the shape of the central filament is specified by the width and the height. A square cross section is assumed, and the thickness is determined by the current. It is supposed that a steel-reinforced copper coil can take 18 MA/m^2 cross section. The divertor coil is plotted with a protective shield of thickness D1SHLD.

For the calculation of the magnetic field, a square number of filaments may be considered, with ND1FIL ranging from 1 to 5. Finally, the location is specified by the x coordinate of the leg with the current flowing in positive z direction. The coil is placed along the x axis . For D1WITH > 0, the leg with the downward current is away from the torus.

/DVC2/ This block is for a two-coil divertor. It is assumed symmetric with the (x,z) plane as a mirror plane. Only the coil with y > 0 is specified explicitly. The meanings of the variables D2WITH, D2HEIT, D2THIK, ND2FIL, and D2SHLD are completely analogous to the one-coil divertor. Note that D2CURT stands for the total current in both coils together. The location of the leg with the current going upwards is at the point with coordinates D2XUP, D2YUP. From this point the leg with the current going down is at the angle D2FI with the positive x axis.

/DVC3/ Here, the data of a three-coil divertor are stored; hence, all the variable names have the prefix D3 or ND3. The three-coil divertor is built up from a one- and a two-coil divertor. The specification of the coils is exactly the same as in the two preceding common blocks. To distinguish the one- from the two-coil part, the last character is replaced by a 1 or a 2, respectively. Hence, D3WIT1 and D3WIT2 designate the widths of the two sorts of coils involved. All the coils are given the same height D3HEIT, the same number of filaments ND3FIL, and the same thickness of the shielding D3SHLD. The total current in all three coils together is D3CURT. The variable D3FRAC specifies the fraction of the total current flowing in the single coil placed along the x axis.

/DVC4/ This block contains the data concerning a four-coil divertor. It is built up from 2 two-coil divertors. The variables group accordingly, and they are distinguished by either a 1 or a 2 in the name.

/DVTC/ Here a generalized T-coil divertor is specified. Like all the other divertors it is symmetric with the (x,z) plane, and only the part in the halfspace $y > 0$ is specified. The starting point is the "knee," which is given by DTX, DTY. The two wings are distinguished by a 1 or a 2 in the variable names. Wing number 1 has the current flowing upward in the vertical part. Each wing is specified by the width DTWIT1, DTWIT2, and the angle with the positive x axis DTF1DG, DTF2DG. For this case, the value of the current DTCURT is the sum of that flowing in any two equivalent wings. The other variables have the same meanings as earlier.

There is no common block for a divertor with more coils. However, the code can handle several divertors simultaneously, and the variables I1CD, I2CD, I3CD, I4CD, and ITCD can assume the values 0,1 independently. To match the separatrix to the plasma edge, the code varies the total current in all the divertor coils. Thereby, the ratio among the currents D1CURT, D2CURT, D3CURT, D4CURT, and DTCURT is held constant if more than one of them is turned on. This makes it necessary to assign some nonzero initial values to all the currents present.

/ROUTE/ This block contains the schedule of a run. All the variables are integers, and each one has a value of either one or zero (IALTER and INDIFL can also be two). Normally, a certain action is taken when they are nonzero.

ILGINP = 1 The subroutine INPUT works in the long input mode.

IALTER = 1 The toroidal field coils closest to the divertor are altered.

IALTER = 2 The toroidal field coils closest to the divertor are replaced by two coils each.

ICLSPS = 1 A drawing of the filaments used in the standard toroidal field coils is made.

ICLSPA = 1 A drawing of the filaments used in the altered toroidal field coils is made.

IRIPL = 1 A ripple plot of the confined field lines is made.

ISHRIP = 1 The ripple plot goes from y = 0 up to the second toroidal field coil.

ISHRIP = 0 The ripple plot goes halfway around the torus.

ICORN = 1 The corners of all the filaments in the toroidal field coils and in the divertor are written onto file FOR12.DAT.

IEDGE = 1	The total current in the divertor is varied so as to connect the separatrix with a predetermined flux surface inside the plasma edge.
I1CD = 1 I2CD = 1 I3CD = 1 I4CD = 1 ITCD = 1	These variables specify which divertors are present. The code also allows more than one variable to be nonzero.
IFIN = 0	The main program will be executed another time with a different configuration.
IFIN = 1	This is the last configuration to be computed.
INDIFL = 0	The field lines are started automatically and are equally spaced in the plane of the second toroidal field coil. The input variable NFL specifies how many field lines on both sides of the separatrix are plotted. The spacing is chosen such that the last field line starts on the magnetic axis.
INDIFL = 1	The starting points of field lines are defined individually. This can be done interactively at the terminal. The code points out the endpoints of the previous field line and asks for a new starting point. The starting point may be either at $y = 0$ (at the divertor) or at $\phi = 3*\pi/NTF$, the location of the second toroidal field coil. A negative value of R terminates the field line plotting.
INDIFL = 2	First, the whole automatic procedure is carried

	out in the same way as with INDIFL = 0. Afterwards, the code asks for further starting points individually, which allows the study of the "private" flux of the divertor.
ISEP = 1	The code looks for a null point of B_ϕ somewhere on the x axis. Then, it plots the separatrix starting from this point.
IBPL = 1	A plot of B_ϕ and $\|B\|$ along the x axis is made. Failure to find a null point automatically sets IBPL = 1.
IVACON = 1	Only the vacuum fields are computed. In this case, the input variable STROM is required. It gives the total current in the toroidal field coils and will also be stored under TFCURT.
IVACON = 0	In this case, the subroutine EQUIN reads the equilibrium data, which also specify the total current TFCURT, the edges of the plasma REDGE1 and REDGE2, and the magnetic axis of the plasma RMA.

/EQUI/ This block contains the equilibrium data. It specifies the functions of $\psi(R,z)$ and $F(\psi)$. The array PSI(65,80) contains values of ψ on an equally spaced (R,z) grid. The possible R and z values are stored in the arrays R(65) and Z(80). The arrays R has NR points over a range RDIM starting at RMIN. The array Z is organized the same way. The common blocks also contain approximations for the derivatives $\partial\psi/\partial R$, $\partial\psi/\partial z$, and $\partial^2\psi/\partial R\partial z$. At the same grid points these derivatives are determined by cubic spline interpolation and are stored in the arrays DPDR(65,80), DPDZ(65,80), D2PDRZ(65,80). Values of the function $F(\psi)$ are stored in the array F(65). The corresponding ψ are in PSF. There are NPSF points over a range DPSF starting at PSFMIN. CUR contains the current in the toroidal field coils. The variables NWDPSI, NWOF count how many times a call is being made to calculate the magnetic field outside the range of tabulated values $\psi(R,z)$ and $F(\psi)$, respectively. The value of the variable INDONE tells if $\psi(R,z)$ and its derivatives have been specified already in a previous call to EQUIN.

/FLPAR/ Here, some parameters of the field lines are stored. The variables DRFL and NFL are used for the automatic launch of the field lines. In case a ripple plot is made, NRP counts the confined field lines. STP, TOL, and DSO are other input variables. STP is used to plot the coils (it gives the step size between grid points in meters); TOL contains the precision requirement for the integrator; and, DSO defines the initial step size for the integrator.

/SEPPAR/ This common block contains some parameters of the separatrix. The variables RMA, REDGE1, REDGE2, and DSCRAP are input variables. DSCRAP gives the thickness of the scrape-off layer. The other three variables define the location of the plasma. In the case of IVACON = 0, they are overwritten by values obtained from the equilibrium data. RMA gives the location of the magnetic axis, and the other two stand for the inner and the outer edge of the plasma, respectively. The major radius, where the separatrix passes under the coil, is stored at the variable RSEPTR. If the code fails to find a null point in B_ϕ, or if the separatrix for some reason should not reach the second toroidal field coil, this value is returned as 999.99.

Finally, the value of EXPANS is a measure for the expansion in volume of the diverted flux. It is the ratio of the distances from the separatrix to the first diverted field line taken at the location of the second toroidal field coil to that at the symmetry plane of the divertor.

The variables XBOMIN, XBOMAX define the interval where the code looks for a zero in B_ϕ when calculating the separatrix. In the case of a simple one-, two-, or three-coil divertor, the code chooses these values itself, but for more complicated configurations, they have to be specified on input.

4. DESCRIPTION OF THE CODE

The main program consists merely of a sequence of calls for subroutines. There is a loop which computes a different configuration each time through. A description of each subroutine is given in the following:

INPUT:	This subroutine reads the data defining the configuration of coils as described earlier.
CHECK:	This subroutine scans the input data for some inconsistencies. If a mistake is detected, the erroneous variables are printed and preceded by the value of the variable IDIAG, which helps to identify the mistake. After finding a mistake, the code continues with the next configuration, if any.
EQUIN:	This routine reads the equilibrium data. As a preparation for later cubic spline interpolation on $\psi(R,z)$, subroutine DERIVS is called to determine the derivatives at the grid points. The flux surface $\psi(R,z) = 0$ is supposed to be the edge of the

plasma, and, therefore, the array PSI is scanned for some change in sign. This is done in the midplane $z = 0$ starting at both ends of the array. Upon detection, the exact null point is determined by the method of half tabling. Intermediate values of ψ are computed by bicubic spline interpolation calling the routines COEFF and BICUBE.

DERIVS:
DERIVD:
COEFF:
BICUBE:

This is a complete package of routines for cubic spline interpolation written by P.W. Gaffney.[1]

SEGPT: This routine fills the common block /WIRES/ with the data concerning the coils. The variable NL counts the total number of current loops; NT counts only the loops in the toroidal field coils. There can be up to 465 different loops, each consisting of up to 6 segments. In the arrays XL, YL, and ZL, the endpoints of these segments are stored. They are listed in the order the current flows, and the first corner is always listed a second time at the end of the loop. The array CF contains the currents in these loops. If the code is to choose a current automatically, so as to match the separatrix to the plasma edge, these values are overwritten by the subroutine SPRTRX. For identification purposes there is also an array NK. The numbers stored here tell which kind of coil a particular loop belongs to.

The coordinates of loops in toroidal field coils are generated by successive rotation of a set of filament loops placed along the x axis. First, these coordinates are determined and stored in the arrays X0, Y0, and Z0. The shape of altered toroidal field coils is described by the arrays XALT1, YALT1, XALT2, and YALT2.

FLINE: This routine plots the field lines. When following the field line, the z component of the magnetic field is ignored and the field line is forced to remain in the (x,y) plane. The resulting plot has a direct physical use only when there exists a set of nested flux surfaces. A field line stays on the same flux surface forever. Ignoring B_z, the plots actually show the intersection of a flux surface with the (x,y) plane.

In any case, a call to DVCPLT is made. This initiates the main plot. Therefore, the thicknesses of the various divertor coils have to be determined.

Depending on the values of ISEP and IEDGE, the separatrix is determined. This involves calls to the subroutine SPRTRX, which requires the knowledge of the divertor currents. In the case of IEDGE = 0, they are specified as input. However, when the code has to match the separatrix to the plasma edge, it executes a procedure to determine the appropriate current. In terms of the total current in the toroidal field coils, the code assumes a fixed interval for allowed divertor currents. First, this interval is scanned in NTF equally spaced steps. Each call to SPRTRX returns a new value of RSEPTR, which determines GAP0 and GAP1. They give the gap between the plasma and the separatrix for the previous and the actual divertor current. If the gap changes sign, the routine starts a finer search for the appropriate current by half tabling. If the scan reaches the maximum current without detection of a negative gap, the search is terminated. After a successful scan, the interval of current leading to positive and negative gap is divided repeatedly until the gap is smaller than 1 cm. Then, SEPPLT is set true, and SPRTRX is called a last time to plot the separatrix. This procedure may not succeed. If a change in current of less than 10^{-4} changes the gap by more than 1 cm, the situation is considered unstable and the routine escapes. In most cases one of the involved values of RSEPTR will be 999.99, which means that there exists no separatrix for that particular current.

After this procedure, the code determines field lines. In the case of INDIFL = 0, they are launched equally spaced in the plane of the second toroidal field coil, with the innermost line being the magnetic axis. There is also an interactive mode to launch the field lines individually. After the field lines are launched, the main plot is terminated, and various auxiliary plots are made.

First, there is the possibility of plotting B_ϕ versus x. If the code failed to find the null point of B_ϕ, this is plotted automatically. Making the plot involves calls to BAX and BAXPL.

Then, there is a ripple plot available. The call to RIPLOT also evaluates the ripple.

The last feature is a plot of the gap versus the total divertor current. It appears only when the code fails to match the separatrix to the plasma edge. If for all of the divertor currents there was no null point in B_ϕ, this plot may show no data points.

SPRTRX: This subroutine determines the separatrix. In this case, the code tries to match the separatrix to the plasma edge. SPRTRX first performs an update of the divertor currents stored in the common block /WIRES/ in the array CF. Then, it determines the null point in B_ϕ. This search is restricted to the interval RMIN, RMAX. In the case of a simple one-, two-, three-coil divertor, some analytical estimates are used. In more complicated cases, however, the code uses the input values XBOMIN, XBOMAX to define the search interval. Then, calls to BAX and XPOINT are made. If they are successful, the flag GOTNIL is set true, and RXPT gives the x coordinate of the null point. Ignoring B_z, this null point has to be on the x axis by symmetry. If the flag indicates a failure, the routine terminates and returns the value RSEPTR = 999.99. Upon successful detection of a null point, calls to FLINT are made to integrate the separatrix. The integration starts at a point slightly displaced from the actual null point to make sure the right part of the separatrix is integrated. In the case SEPPLT = FALSE, only the part passing through the toroidal field coils is integrated in order to determine RSEPTR. No plots are made. When SEPPLT is set true, however, both parts of the separatrix are integrated and plotted.

BAX: This little routine fills the common block /BAXVAL/ with data concerning the magnetic field on the x axis in the interval RMIN, RMAX.

XPOINT: This routine determines the null point in B_ϕ. First, it scans the array BFI in common block /BAXVAL/ for a change in sign. If it doesn't find any, it returns with the flag GOTMIN set false. If it does find two values in BFI with opposite signs, the interval of corresponding values in R is divided subsequently, and the magnetic field is evaluated.

The procedure stops when the null point is determined to an accuracy of 0.1 mm.

FLINT: This subroutine performs the actual integration of field lines. It works in Cartesian coordinates and solves the differential equations:

$$\frac{dx}{ds} = \frac{B_x}{B} ,$$

$$\frac{dy}{ds} = \frac{B_y}{B} ,$$

$$\frac{dz}{ds} = \frac{B_z}{B} .$$

These equations are evaluated by the subroutine EQFL, which is called by the integrator but which has to be declared here. Before starting the integration, FLINT determines the sign of the increment DS so as to launch the field line in the right direction. Then, it enters a loop of subsequent calls to the integrator STEP. There are a number of different escapes from this loop.

After each step, FLINT stores the coordinates of the new point, the azimuth angle ϕ, and $|B|$ in the arrays XFL, YFL, FIFL, and BMOD, respectively. If they are about to overflow, integration is stopped, and the curve calculated so far is plotted.

Should the integrator be unable to maintain the required precision, the call is tried again, which results in another order of approximation. After the third recall, FLINT escapes and plots what has been calculated so far. This occurs very rarely, perhaps only when launching a field line with an unreasonable precision requirement.

Next, FLINT checks to see if the field line becomes caught in a spiral. Therefore, the angle between subsequent field line segments is monitored. It is evaluated from the cross product of subsequent segment vectors. These incremental turning angles are summed up and stored at SPIR. If the sum of the angles exceeds $4*\pi$, FLINT escapes. This phenomenon may happen when the field line comes too close to a wire. However, it is observed only under slightly different circumstances. On a symmetry plane of the toroidal field coil configuration outside the torus, many large contributions from all the coil segments cancel. This creates the danger of large round-off errors, and a field line may continue in a wrong direction after hitting such a region (see Fig. 2).

A possible reaction to such a mishap may be to increase the precision requirement. Consequently, the integrator reduces the step size, and, eventually, the code will not be able to pass through the dangerous zone because the storage capacity is exceeded. However, there is a more profound problem associated with this reaction. Given the precision requirement and the machine precision, the integrator determines the optimal step size. The function subroutine providing the differential equation is assumed to be absolutely precise. Ignoring these errors, it may choose an absurdly small increment DS. The subroutine FLINT checks if $|DB| < TOL*|B|$. If this happens three consecutive times, it escapes.

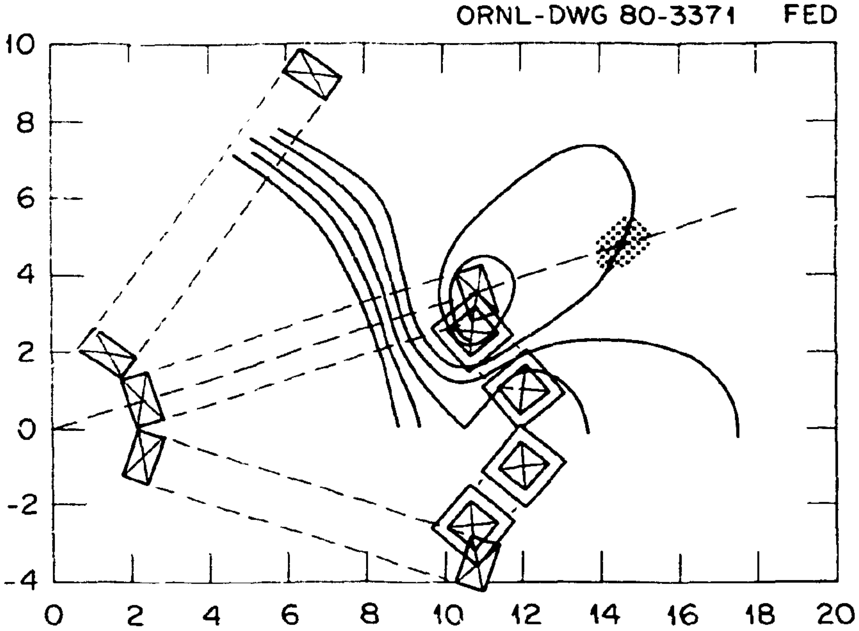


Fig. 2. On the symmetry lines outside the toroidal field coils, the error in calculating the field becomes large, and the field line integration may lead into a wrong direction.

After passing all these tests, the code finds out if the anticipated end of the field line has been reached. It checks to see if one of the lines FI = FIMAX or FI = 0 has been crossed. The action taken in such a case depends on a number of things. Normally, the field line is plotted, and the data necessary for the ripple plot are written onto file FOR08.DAT. If the field line was launched at FI = FIMAX, and if a ripple plot halfway around the torus is required, the field line is restarted in a backward direction when reaching FI = 0.

BSYM: This routine provides the symmetrical field and derivatives due to a current along the z axis.

BEQUI: This routine calculates the equilibrium field. It interpolates tables of the functions $\psi(R,z)$ and $F(\psi)$, which are stored in the common block /EQUI/. These functions are solutions of the Grad-Shafranov equation, and the fields and partial derivatives are calculated by the formulas,

$$B_r = -\frac{1}{R}\frac{\partial\psi}{\partial z} ,$$

$$B_\phi = -\frac{1}{R} F(\psi) ,$$

$$B_z = \frac{1}{R}\frac{\partial\psi}{\partial R} ,$$

$$\frac{\partial B_r}{\partial R} = -\frac{B_r}{R} - \frac{1}{R}\frac{\partial^2\psi}{\partial R\partial z} ,$$

$$\frac{\partial B_r}{\partial \phi} = 0 ,$$

$$\frac{\partial B_r}{\partial z} = - \frac{1}{R} \frac{\partial^2 \psi}{\partial z^2} ,$$

$$\frac{\partial B_\phi}{\partial R} = - \frac{B_\phi}{R} - \frac{1}{R} \frac{dF}{d\psi} \frac{\partial \psi}{\partial R} ,$$

$$\frac{\partial B_\phi}{\partial \phi} = 0 ,$$

$$\frac{\partial B_\phi}{\partial z} = - \frac{1}{R} \frac{dF}{d\psi} \frac{\partial \psi}{\partial z} ,$$

$$\frac{\partial B_z}{\partial R} = - \frac{B_z}{R} + \frac{1}{R} \frac{\partial^2 \psi}{\partial R^2} ,$$

$$\frac{\partial B_z}{\partial \phi} = 0 ,$$

$$\frac{\partial B_z}{\partial_z} = \frac{1}{R} \frac{\partial^2 \psi}{\partial R \partial z} .$$

To interpolate the table $\psi(R,z)$, calls to COEFF and BICUBE are made. $F(\psi)$ is interpolated by third-order Lagrange interpolation. Subroutine LAGR also evaluates the derivative. If the field point is outside the table $\psi(R,z)$, a call to BSYM is made, and the vacuum field is returned. Then, $F(\psi)$ is interpolated, calling LAGR. If the interpolated value $\psi(R,z)$ is positive, the field point lies outside the plasma, and $F(\psi)$ assumes the value at the edge of the plasma. If, by some slight interpolation error, ψ should come out less than the value on the magnetic axis, the axis value is used.

BFIELD: This routine calculates an approximation for the magnetic field. The divertor is considered as a perturbation to the equilibrium field in the symmetrical torus, as is the ripple field due to the discreteness of the toroidal field coils:

$$\mathrm{BF} = \mathrm{BO} + \mathrm{BV} - \mathrm{BS} \ ,$$

where

BQ = field with the plasma in a two-dimensional, symmetrical equilibrium;

BV = vacuum field due to all the toroidal field coils and the divertor; and

BS = symmetrical vaccuum field due to the total current in all the toroidal field coils flowing along the z axis.

The common block /FIELD/ is the memory of this routine. It checks if a particular field point has been calculated previously so as to avoid unnecessary computation.

The calculation of the vacuum field BV involves calls to BWIREC for each of the wire segments described by the common block /WIRES/. If the input variable IVACON is set at one, these are the only computations performed. The other fields, BQ and BS, keep their zero initial values in this case. If IVACON = 0, calls to BSYM and BEOUI are made, and the resulting field is calculated as described above. Finally, in order to restrict the field line to the (x,y) plane, the z components are set to zero.

BWIREC: This subroutine evaluates the magnetic field and derivatives due to a current carrying line segment. It is taken out of the FLOC code and described elsewhere.[2]

STEP: This routine integrates a set of first-order differential equations step by step. (It is published in Ref. 3.)

MACHIN: This routine evaluates the machine precision for the integrator.

EQFL: This routine computes the differential equations for the integrating routine.

CYLCAR: This routine performs a transformation from cylindrical to Cartesian coordinates. It transforms both the magnetic field vector and the derivatives.

LAGR: This routine performs a third-order Lagrange interpolation according to the formula

$$\begin{aligned} f(x) = {} & \frac{(x - x_j)(x - x_{j+1})(x - x_{j+2})}{(x_{j-1} - x_j)(x_{j-1} - x_{j+1})(x_{j-1} - x_{j+2})} f(x_{j-1}) \\ & + \frac{(x - x_{j-1})(x - x_{j+1})(x - x_{j+2})}{(x_j - x_{j-1})(x_j - x_{j+1})(x_j - x_{j+2})} f(x_j) \\ & + \frac{(x - x_{j-1})(x - x_j)(x - x_{j+2})}{(x_{j+1} - x_{j-1})(x_{j+1} - x_j)(x_{j+1} - x_{j+2})} f(x_{j+1}) \\ & + \frac{(x - x_{j-1})(x - x_j)(x - x_{j+1})}{(x_{j+2} - x_{j-1})(x_{j+2} - x_j)(x_{j+2} - x_{j+1})} f(x_{j+2}) \ . \end{aligned}$$

The subroutine also evaluates the derivative of this formula. In order to cover the entire range of arguments, three additional values are added at both ends of the table. They are chosen so as to simulate a maximum at the plasma edge and a minimum on the axis. This choice takes into account the fact that $dF/d\psi = 0$ at both of these places.

OUTDAT: This routine lists the major parameters of a coil configuration in the form of a plot. On request, it also writes the data in the common block /WIRES/ onto file FOR12.DAT.

BAXPL: This routine makes a plot of B_ϕ versus x. It merely uses the data stored in the common block /BAXVAL/.

RIPLOT: This routine reads for each field line the values of ϕ and B_ϕ from the file FOR08.DAT. It determines the ripple for all the lines, chooses appropriate scales, and makes a plot of B_ϕ versus ϕ.

TFCPLT:
DVCPLT:
TFDRAW:
DVDRAW:
DTDRAW:
SYMBO:
SYMB7:
TRANS:

This whole package of subroutines serves to plot the configuration of coils. The routines TFDRAW, DVDRAW, and DTDRAW plot one single coil situated at $\phi = 0$. The routines TFCPLT and DVCPLT repeatedly call these routines and, each time, specify the appropriate transformation. The transformation is performed by TRANS.

5. SOME RESULTS

5.1 TWO-COIL DIVERTOR

In Fig. 3 a simple two-coil divertor is shown. Because the diverted flux is directed into a region of enhanced field strength, the expansion is poor. In order to achieve better expansion, we might reduce the size of the toroidal field coils as shown in Fig. 4. However, although the current in the altered toroidal field coil is chosen so as to minimize the ripple, this configuration produces more ripple than the same divertor without the altered toroidal field coils.

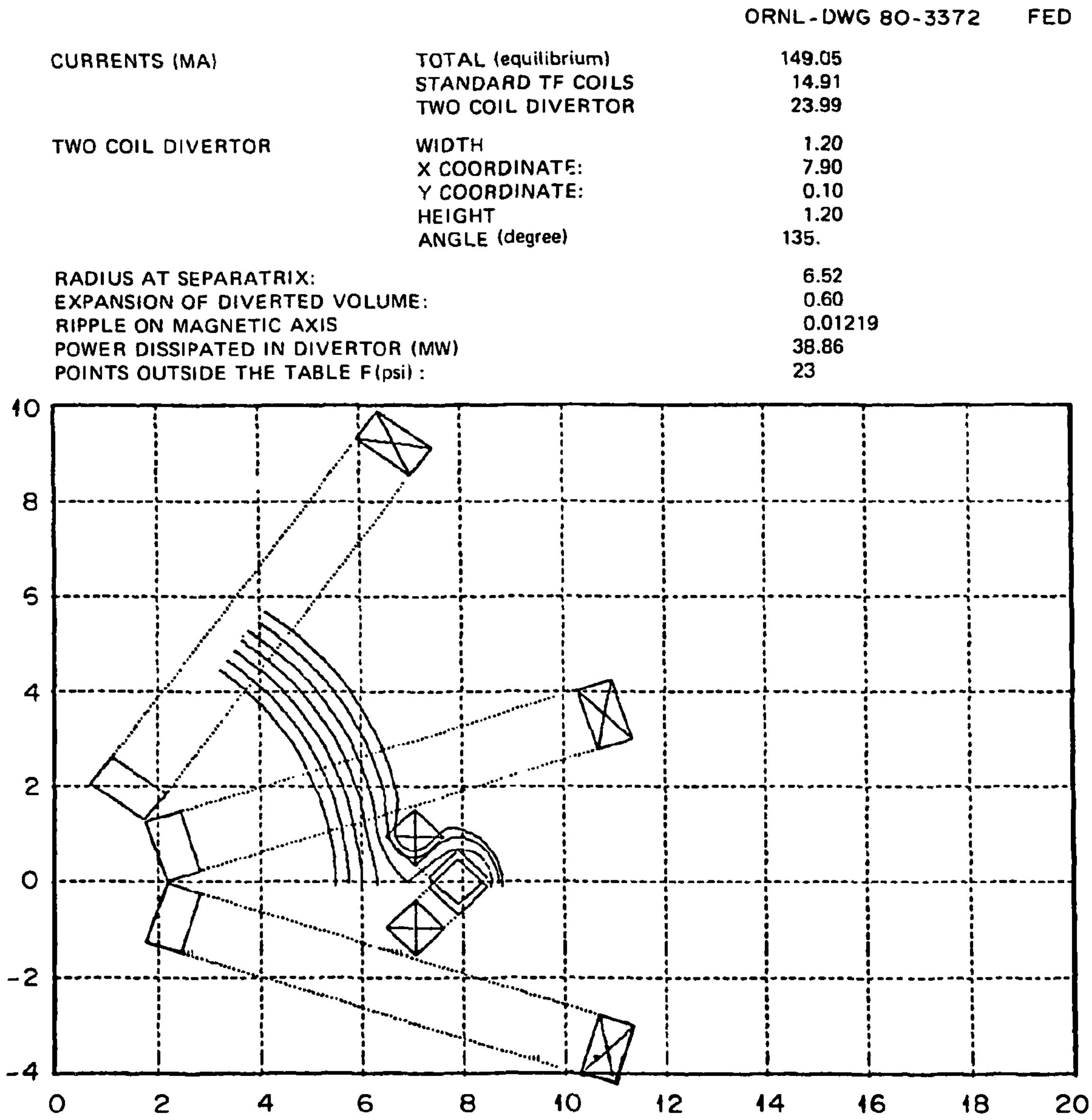
The two figures have been chosen to illustrate two rules for the design of good divertors:

(1) In order to achieve good expansion, the diverted flux must be directed into a region of lower field strength.

(2) A small divertor close to the plasma keeps the ripple down.

There are two physical reasons for the second rule. First, the smaller the divertor coils are, the closer the far field begins. In the far field, the field strength drops off quickly, like some high order multipole. In the near field, however, the closest wire dominates, and the drop-off is basically like 1/r. Therefore, one might wish to have the magnetic axis in the far field of the divertor in order to reduce the perturbing field in the bulk of the plasma. Second, there is the simple fact that a small coil produces a higher field at its center than a big coil with the same current. Therefore, a small divertor uses the current more efficiently for the cancellation of the toroidal field.

It must be pointed out, however, that a proper divertor must have a certain minimum size in order to divert a sufficiently thick layer of plasma. So, in the following examples, the requirement is adopted that neither the separatrix nor the first diverted field line should hit the divertor coils. In the unperturbed plasma these two field lines are



CURRENTS (MA)	TOTAL (equilibrium)	149.05
	STANDARD TF COILS	14.91
	TWO COIL DIVERTOR	23.99
TWO COIL DIVERTOR	WIDTH	1.20
	X COORDINATE:	7.90
	Y COORDINATE:	0.10
	HEIGHT	1.20
	ANGLE (degree)	135.
RADIUS AT SEPARATRIX:		6.52
EXPANSION OF DIVERTED VOLUME:		0.60
RIPPLE ON MAGNETIC AXIS		0.01219
POWER DISSIPATED IN DIVERTOR (MW)		38.86
POINTS OUTSIDE THE TABLE F(psi) :		23

Fig. 3. With simple two-coil divertors, ripple values of about 1% can be achieved, but such divertors are too small to push the diverted field lines all the way to the outside of the toroidal field coils.

ORNL-DWG 80-3373 FED

CURRENTS (MA)	TOTAL (equilibrium)	149.05
	STANDARD TF COILS	14.91
	FIRST ALTERED TF COILS	8.50
	TWO COIL DIVERTOR	24.34
1ST ALTERED TF COILS	WIDTH	5.60
	ANGLE (degree)	15.0
TWO COIL DIVERTOR	WIDTH	2.50
	X COORDINATE:	9.00
	Y COORDINATE:	0.20
	HEIGHT	2.00
	ANGLE (degree)	135.
RADIUS AT SEPARATRIX:		6.52
EXPANSION OF DIVERTED VOLUME:		3.09
RIPPLE ON MAGNETIC AXIS		0.04917
POWER DISSIPATED IN DIVERTOR (MW)		78.85
POINTS OUTSIDE THE TABLE PSI(R, Z):		108
POINTS OUTSIDE THE TABLE F(psi) :		132

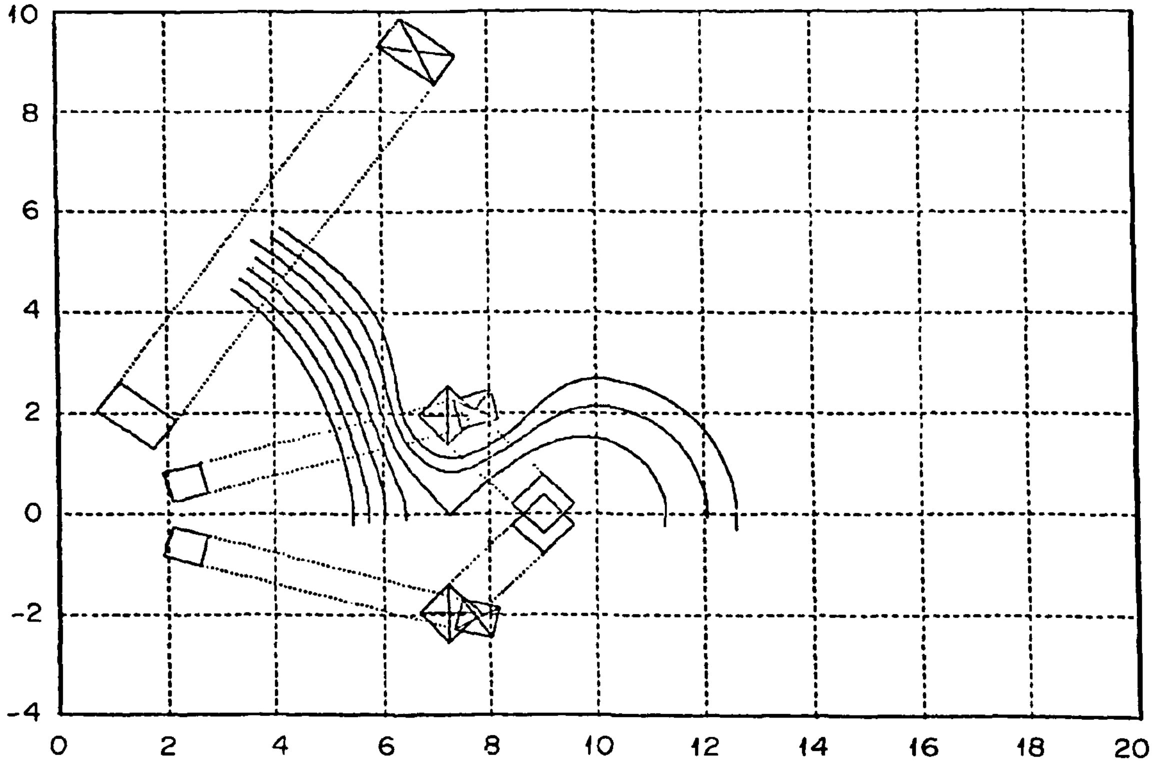


Fig. 4. A two-coil divertor can achieve good expansion of the diverted flux, but the ripple of such a configuration is higher than without changing the toroidal field coils.

about 30 cm apart. Perhaps even this is too weak a requirement because it ignores the need of shielding for the divertor coils.

5.2 T-COIL DIVERTOR

T-coil divertors produce somewhat less ripple than two-coil divertors of the same size. This is because the radial horizontal parts contribute more efficiently to the cancellation of the toroidal field. As an example, a T-coil divertor producing 0.794% ripple has been found that diverts a plasma layer ~25 cm thick (see Fig. 5). If we settled for a thinner diverted plasma layer, surely a smaller T-coil divertor could be found closer to the plasma that would produce even less ripple.

5.3 TUBE DIVERTOR

In order to bring the flux diverted by a T-coil configuration away from the torus, one might pass the diverted field lines through a whole sequence of coils (see Fig. 6). This action has the disadvantage of dissipating much power in all the divertor coils (>100 MW). Furthermore, diffusion across the field lines might become an important factor, and too many particles might get lost on their long way through this complicated divertor. Through a careful choice of each pair of added coils, the ripple is still determined by the T-shaped coils in front.

ORNL-DWG 80-3374 FED

CURRENTS (MA)	TOTAL (equilibrium)	149.05
	STANDARD TF COILS	14.91
	T-COIL DIVERTOR	14.67
T-COIL DIVERTOR	X COORDINATE:	7.05
	Y COORDINATE:	0.01
	LENGTH OF FIRST WINGS	0.85
	ANGLE (degree)	0.0
	LENGTHS OF SECOND WINGS	0.85
	ANGLE (degree)	90.0
	HEIGHT	1.15
RADIUS AT SEPARATRIX:		6.52
EXPANSION OF DIVERTED VOLUME:		0.09
RIPPLE ON MAGNETIC AXIS		0.00794
POWER DISSIPATED IN DIVERTOR (MW)		29.05
POINTS OUTSIDE THE TABLE F(psi):		22

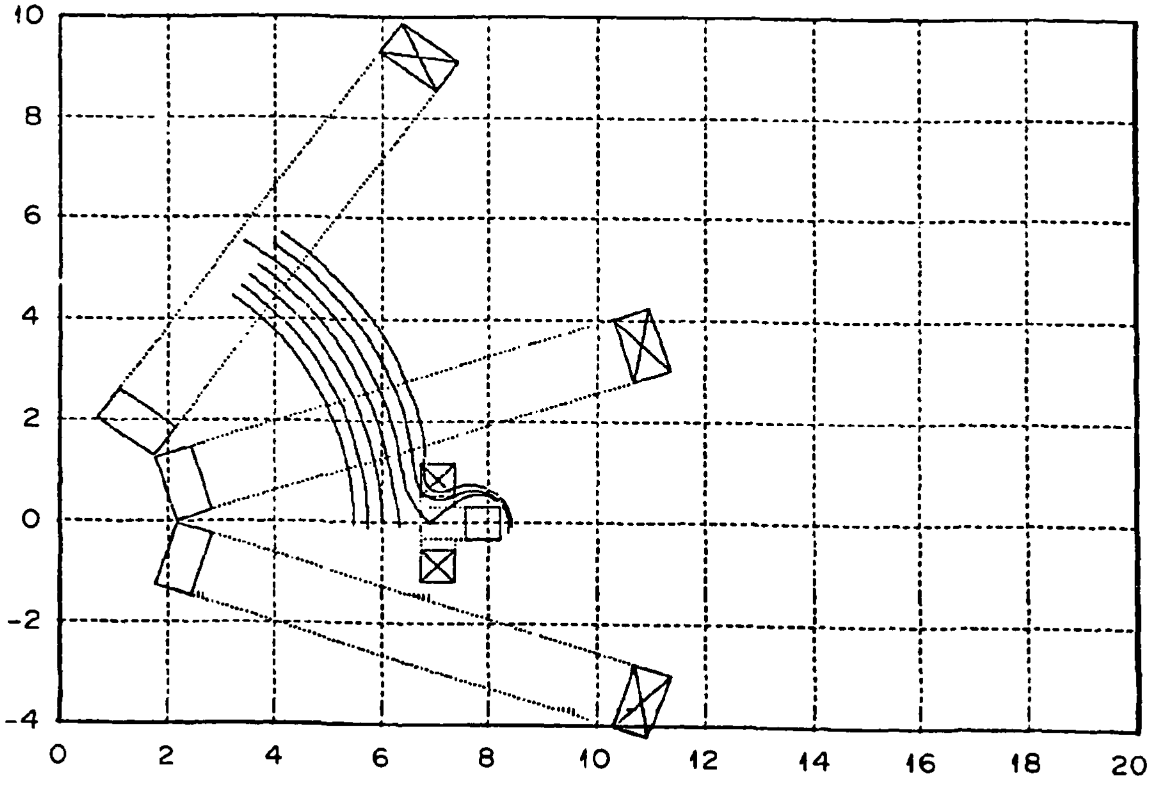


Fig. 5. A T-coil divertor producing 0.794% ripple. It diverts a plasma layer ~25 cm thick.

ORNL-DWG 80-3375 FED

CURRENTS (MA)	TOTAL (equilibrium)	149 05
	STANDARD TF COILS	14 91
	TWO COIL DIVERTOR	14 68
	THREE COIL DIVERTOR	11 16
	FOUR COIL DIVERTOR	29 36
	T-COIL DIVERTOR	14 68
TWO COIL DIVERTOR	WIDTH	1 20
	X COORDINATE	9.00
	Y COORDINATE	0 10
	HEIGHT	1 20
	ANGLE (degree)	115
THREE COIL DIVERTOR	HEIGHT	2 00
	WIDTH OF PAIR COILS	2 00
	X COORDINATE	13 00
	Y COORDINATE	0 20
	ANGLE (degree) OF PAIR COILS	150
	WIDTH OF SINGLE COIL	3 00
	X COORDINATE	100 00
	FRACTION OF CURRENT	0 010
FOUR COIL DIVERTOR	FIRST PAIR	
	X COORDINATE	10 00
	Y COORDINATE	0 10
	ANGLE (degree)	115.0
	WIDTH	1 20
	HEIGHT	1.20
	FRACTION OF CURRENT	0 500
	SECOND PAIR	
	X COORDINATE	11.00
	Y COORDINATE	0 10
	ANGLE (degree)	115 0
	WIDTH	1 20
	HEIGHT	1 20
T-COIL DIVERTOR	X COORDINATE	7 05
	Y COORDINATE	0.01
	LENGTH OF FIRST WINGS	0.85
	ANGLE (degree)	0 0
	LENGTHS OF SECOND WINGS	0 85
	ANGLE (degree)	90 0
	HEIGHT	1 15
RADIUS AT SEPARATRIX		6 52
EXPANSION OF DIVERTED VOLUME		1.97
RIPPLE ON MAGNETIC AXIS		0.00942
POWER DISSIPATED IN DIVERTOR (MW)		132.60
POINTS OUTSIDE THE TABLE PSI(R, Z)		331
POINTS OUTSIDE THE TABLE F(psi)		348

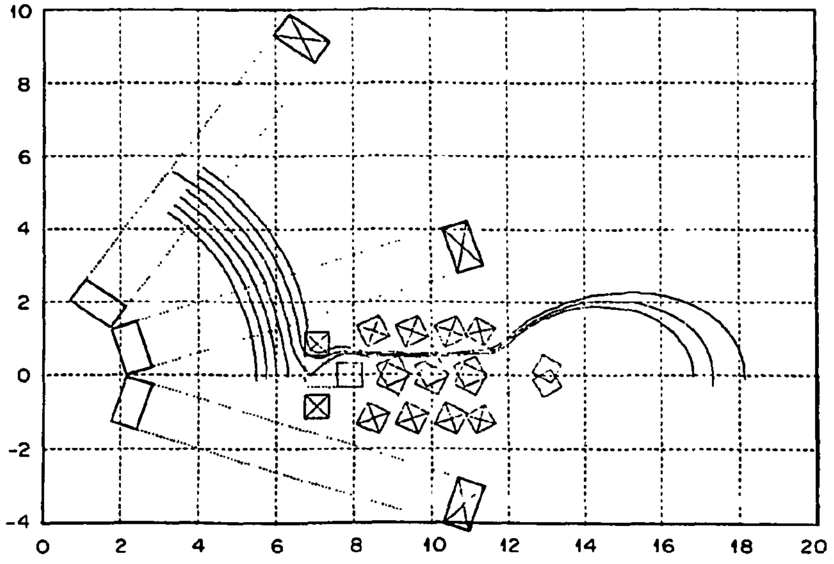


Fig. 6. Using a whole sequence of pairs of coils, the flux diverted by T-coils can be brought out of the toroidal field without a significant increase of the ripple.

REFERENCES

1. P. W. Gaffney, Fortran Subroutines for Bicubic Spline Interpolation, Oak Ridge National Laboratory Report ORNL/CSD/TM-67, (July 1979).

2. R. H. Fowler et al., FLOC — Field Line and Orbit Code for the Study of Ripple Beam Injection into Tokamaks, Oak Ridge National Laboratory Report ORNL/TM-6293 (June 1978).

3. L. F. Shampine and M. K. Gordon, Computer Solution of Ordinary Differential Equations, W. H. Freeman Co., San Francisco, 1975.