Fast Control of Plasma Surface

W.Feneberg,
K.Lackner, P.Martin

IPP 1/223 November 1983

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK
8046 GARCHING BEI MÜNCHEN

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

Fast Control of Plasma Surface

W.Feneberg,
K.Lackner, P.Martin

IPP 1/223 November 1983

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

Fast Control of Plasma Surface

W.Feneberg, K.Lackner, P.Martin

Abstract

We describe a fast method for the identification of the plasma boundary using results of magnetic measurements. The poloidal flux equation is solved inside the region bounded by the location of flux loops using the measured data as boundary values. Additional currents, substituting the plasma currents are distributed over a second control surface, chosen to lie inside the supposed plasma region. This surface current distribution is parameterized in the form of a truncated Fourier series whose coefficients are determined so as to yield an optimum fit to magnetic field component measurements by pick-up coils.

By separating the inner region from all outside effects, this method avoids the difficulties arising in calculating the effects of the iron core present in JET.

Contents page

1. Introduction

For a fast identification of the plasma boundary in tokamaks, the unknown plasma current distribution is usually substituted by several current filaments at fixed positions, where the single wire currents are adjusted so as to give an optimum fit to magnetic signals measured outside the plasma. A particular variant of this method is applied at ASDEX, where a single wire is used, but both its current and position are treated as variable.

The general way of expressing the fields of the single wire currents is the use of Green's function (i.e. Biot Savart's law). This direct way is suitable for Tokamaks with an air transformer, where also all other contributions to the poloidal field are assumed to be known and expressed in a similar fashion.

This method becomes uncertain and leads to calculations with long computer time (Blum-Code /1/), if as in JET the induction of currents is changed by the presence of an iron transformer having a complicated shape. Therefore in this paper we follow an alternative approach, which is possible due to JET's feature of providing knowledge of the flux functions along a closed surface C surrounding the plasma, in addition to measurement of the (tangential) poloidal field component on a second, nearby surface (Fig. 1). Assuming the two surfaces as coincident and the data known on them everywhere, this would constitute a Cauchy-type initial value problem for the elliptic equation

$$\Delta^* \psi = \frac{\partial^2 \psi}{\partial R^2} - \frac{1}{R}\frac{\partial \psi}{\partial R} + \frac{\partial^2 \psi}{\partial z^2} = 0$$

over the vacuum region between the surface C and the plasma boundary. Formulated like this, however, the problem is not well posed in the sense of Hadamard, and requires a regularization procedure for resulting in a numerically stable algorithm.

The present approach - like all the so-called wire codes - starts from the assumption that in the vacuum region, up to the plasma boundary,the fields produced by plasma currents can be adequately approximated by those produced by adjustable currents at fixed positions. This would be rigorously true, if their assumed location would be a closed shell coincident with the plasma surface on which continuous surface currents were allowed to flow. Even in that case however, the determination of these surface currents from flux and field values measured at distant locations would be an ill-posed problem. This is also intuitively clear, as higher moments of the plasma or surface current distribution produce fields decaying rapidly outward and can therefore, conversely, not be determined from such data in a numerically stable way.

The solution to this consist in eliminating such higher moments from the current distribution to be fitted,by either allowing only currents in few discrete locations ("wires") or by assuming a more continuous distribution represented however by a Fourier series truncated at a low mode number.(The effect of the latter procedure can also be obtained by a regularization method penalizing large alternating currents at near-by locations.)

Truncation of the current distributions to be fitted is also consistent with our physical intentions as we aim at determining only the total current, the radial and vertical position, and the elongation and triangularity of the plasma column. Having obtained a substitute plasma current distribution fitting the measured signals, the plasma boundary is identified with the innermost flux surface tangent to a given limiter contour.

We have tested two different models for the substitute plasma currents. In a first model we approximate them by a certain number of wire loops in fixed positions and determine the current strength by some minimum square deviation principle from the magnetic probe measurements. In a second model we choose an arbitrary surface F (the so-called control surface) situated inside the region bounded by C,and distribute currents on it continuously in Fourier modes. The different modes (m = 0, 1, 2, 3, ...) correspond to circular, elliptic, triangular a.s.o. shape of the plasma boundary. The best results have been obtained for mode numbers between 3 and 4 using the same minimum principle to define the current strength as in the wire loops model.

The second model has - compared to the model with wire loops - the advantage that the plasma surface varies only very weakly with a shift or a deformation of the control surface whereas a displacement of the wire loops deforms the surface more strongly. Because of the better stability of the plasma surface we use only the method of the control surface in the plasma identification code and give the formulas in chapter 2 only for this case.

The "surface currents" i_c which describe the external part in the magnetic flux have been expanded in Fourier modes on C. The boundary conditions have been fulfilled at the positions of the flux loops. This gives a system of M linear equations for the M Fourier modes (2M = 14 = number of the flux loops). The matrix of the system and the inverse matrix are geometrical quantities which can be precalculated. In this way the computational effort for the determination of the substitute current distribution is reduced to three multiplications of vectors with matrices (where the number of rows and column are given by the number of pick up coils/flux loops and the Fourier modes used for their analysis, respectively) and the solution of <u>one</u> linear system of equation for the Fourier coefficients of the surface currents on the inner contour. The major part of the computation time is spent for the subsequent tracking of the plasma boundary: if only 6 "radial" grid points are used as basis for the latter process on each ray, the total CPU time needed on a Cray is 7 msec.

2. Mathematical Description

Fig. 1 shows the array of flux loops, pick up coils and a control surface in JET dimensions. A toroidal wire loop at R_s, Z_s with current I produces the flux ψ (R, Z) in R , Z :

$$\psi(R, Z) = \frac{\mu_0}{\pi} I A$$

(All units are in ISU)

A (R, Z, R_s, Z_s) is the Green function /2/

$$A = \frac{\sqrt{R R_s}}{k}\left(\left(1 - \frac{1}{2}k^2\right)K(k) - E(k)\right) \quad \text{- - - - (1)}$$

$$k^2 = \frac{4 R R_s}{(R + R_s)^2 + (Z - Z_s)^2} \quad \text{- - - - (2)}$$

E (k) and K (k) are the elliptic integrals of the first and second kind.

The magnetic field

$$B_R = -\frac{1}{R}\frac{\partial \psi}{\partial Z} \qquad B_Z = \frac{1}{R}\frac{\partial \psi}{\partial R}$$

is given by:

$$B_R = \frac{\mu_0}{\pi} I D_Z \qquad B_Z = \frac{\mu_0}{\pi} I D_R \quad \text{- - - - (3)}$$

$$D_R = \frac{(Z - Z_s)k}{4R(R R_s)^{1/2}}\left(-K(k) + \frac{R^2 + R_s^2 + (Z - Z_s)^2}{(R - R_s)^2 + (Z - Z_s)^2}E(k)\right) \quad (4)$$

$$D_z = \frac{k}{4(R R_S)^{1/2}} \left(K(k) + \frac{R_S^2 - R^2 - (z - z_S)^2}{(R - R_S)^2 + (z - z_S)^2} E(k) \right) \quad \cdots (5)$$

We proceed from the usual equation for the poloidal flux in terms of a given toroidal current density distribution

$$\Delta^* \psi = - \mu_0 R i_p$$

$$\Delta^* = \frac{\partial^2}{\partial z^2} + \frac{\partial^2}{\partial R^2} - \frac{1}{R} \frac{\partial}{\partial R} \quad \cdots\cdots (6)$$

and split the total flux function ψ into two parts

$$\psi = \psi_p + \psi_{ex},$$

where ψ_p is the contribution due to currents subsituting the true plasma currents

$$\Delta^* \psi_p = - \mu_0 R i_p \quad \cdots\cdots (7)$$

and ψ_{ex} contains the flux due to currents flowing outside the considered region.

$$\Delta^* \psi_{ex} = 0 \quad \cdots\cdots (8)$$

We are looking for solutions of (7) and (8) inside a region limited by the boundary surface C of the flux loops (see Fig. 1). For this purpose we use the Green-function applied on currents flowing in toroidal direction on the surface C of the flux loops:

$$i(C) = \sum_{m=1}^{M_1} a_m \cos(m-1)\vartheta + \sum_{m=1}^{M_2} d_m \sin m\vartheta \quad \cdots (9)$$

(ϑ is the poloidal angle in a system of polar coordinates fixed at R_o.)

This ansatz corresponds to the common case where the plasma surface shows no symmetry to the midplane. If $\Psi(R_R, Z_k)$ is the measured flux in the flux loop at position R_R, Z_k, then we split up the quantities in $\frac{K}{2} + 1$ symmetric boundary values

$$\Psi_s(R_k, Z_k) = \frac{1}{2}\left(\Psi(R_k, Z_k) + \Psi(R_k, -Z_k)\right), \; k = 1, 2, \ldots, \frac{K}{2}+1 \qquad (10)$$

and in $\frac{K}{2} - 1$ antisymmetric boundary values

$$\Psi_a(R_k, Z_k) = \frac{1}{2}\left(\Psi(R_k, Z_k) - \Psi(R_k, -Z_k)\right); \; k = 1, 2, \ldots, \frac{K}{2}-1 \qquad (11)$$

with k = 14 the number of flux loops.

From the symmetric part we can calculate $M_1 = \frac{K}{2} + 1 = 8$ cos-modes, but from the antisymmetric part only $M_2 = \frac{K}{2} - 1 = 6$ sin-modes. The reason for this disadvantage in the determination of the sin-modes is due to the fact, that loops Nr. 1 and 8 have a position in the midplane.

In the same way the measured component of the magnetic field b in the pick up coils will be separated into $\frac{K'}{2}$ symmetric (b_s) and $\frac{K'}{2}$ antisymmetric (b_a) quantities, with K = 18 the number of pick up coils. (Here the number of the symmetric and antisymmetric parts are the same, because none of the pick up coils is fixed in the mid plane.)

The plasma currents will be simulated by a continuous distribution on the control surface

$$i_p(F) = \sum_{m'=1}^{M'} b_{m'} \cos(m'-1)\vartheta + C_{m'} \sin(m'-1)\vartheta \quad \ldots. \qquad (12)$$

The mode number M' we choose always much smaller than the maximum number $\frac{K}{2}$ of the pick up coils $M' \ll \frac{K'}{2}$, otherwise we would get results, which show high oscillations in the plasma surface and are therefore useless.

One observes, that we have two systems of equivalent equations, one for Ψ_s, b_s and the other for Ψ_a, b_a. Therefore in the following only the equations for the symmetric, measured quantities Ψ_s, and b_s will be given. Only at the end, when we calculate the total flux and the total magnetic field at a grid point, we collect the symmetric and antisymmetric contributions. The fluxes $\Psi_{ex}(R_g, Z_g)$ and Ψ_p (R_g, Z_g) at a given grid point R_g, Z_g, produced by the surface currents $i(C)$, and the substitute plasma currents $i_p'(F)$, respectively, are given by

$$\Psi_{ex} = \frac{\mu_0}{\pi} \sum_{m=1}^{M_1} a_m B_{g,m} \qquad (13)$$

$$\Psi_p = \frac{\mu_0}{\pi} \sum_{m'=1}^{M'} b_{m'} B^*_{g,m'}$$

with

$$B_{g,m} = \int_C \cos(m-1)\vartheta \, A(R_g, Z_g, R_c, Z_c) \, ds \qquad (14)$$

and

$$B^*_{g,m'} = \int_F \cos(m'-1)\vartheta \, A(R_g, Z_g, R_f, Z_f) \, ds \qquad (15)$$

The geometrical quantities $B_{g,m}$, $B^*_{g,m'}$ are precalculated for all grid points and at the positions R_k, Z_k of the flux loops. If we write $\Psi_{ex} = \Psi_1 + \Psi_2$, then we have on the surface C the boundary conditions:

$$\Psi_1(R_k, Z_k) = \frac{\mu_0}{\pi} \sum_{m=1}^{M_1} a_m B_{k,m} \qquad (16)$$

$$= \Psi_{\text{measured}}$$

and

$$\Psi_2(R_k, z_k) = -\frac{\mu_0}{\pi}\sum_{m'=1}^{M'} b_{m'} B^{*}_{k,m'} \quad \text{------} \quad (17)$$

This splitting-up of the surface currents on C is convenient as Ψ_1 will depend therefore only on flux loop measurements. The pick-up coils signals are to be fitted by the flux combination $\Psi_2 + \Psi_p$, which in turn satisfies homogeneous boundary conditions along C. (A coupling between the two distributions arises, as Ψ_1 will also contribute to the magnetic field at the location of the pick-up coils; this contribution is conveniently evaluated and subtracted from the measured signal before the determination of $\Psi_2 + \Psi_p$.)

Equation (16) can be solved by Matrix inversion

$$a^1_m = \frac{\pi}{\mu_0}\Psi_s(R_k, z_k) B^{-1}_{k,m} \quad \text{----} \quad (18)$$

whereby $B^{-1}_{k,m}$ has been precalculated. To prepare the solution of equ. (17) we use again the linearity of the differential equation for Ψ_2, $\Delta^*\Psi_2 = 0$, by the assumption:

$$\Psi_2 = -\frac{\mu_0}{\pi}\sum_{m'=1}^{M'} b_{m'}\hat{\Psi}_{2,m'}, \quad \text{----} \quad (19)$$

where $\hat{\Psi}_{2,m'}$ is determined by $\Delta^*\hat{\Psi}_{2,m'} = 0$ and the boundary condition

$$\hat{\Psi}_{2,m'}(R_k, z_k) = B_{k,m'} \quad \text{----} \quad (20)$$

$\hat{\Psi}_{2,m'}$ plays the role of the fundamental solution, which can be found in analogy to equ. (18):

$$\hat{\Psi}_{2,m'}(R_k, z_k) = B_{k,m'} = \sum_{m=1}^{M_1} \alpha_{m,m'} B_{k,m} \quad \text{---} \quad (21)$$

$$\alpha_{m,m'} = B_{k,m'} \times B^{-1}_{k,m} \quad \text{----} \quad (22)$$

Using this precalculated quantity and equ. (18) we find the total flux $\Psi_S(R_g, Z_g)$ for each given grid point R_g, Z_g in the volume bounded by C and the control surface F

$$\Psi_S(R_g, Z_g) = \frac{\mu_0}{\pi}\sum_{m=1}^{M_1} a_m^1 B_{g,m} + \frac{\mu_0}{\pi}\sum_{m'=1}^{M'} b_{m'} B_{g,m'} - \frac{\mu_0}{\pi}\sum_{m'=1}^{M'} b_{m'}\left(\sum_{m=1}^{M_1} \alpha_{m,m'} B_{g,m}\right) \quad (23)$$

We will now proceed to calculate the components $B_{R,calculated}$ and $B_{Z,calculated}$ at the positions of the pick up coils R_i, Z_i from equ. (23) using the derivatives (4) and (5) of the Green function.

These are given by

$$B_{R,calculated}(R_i, Z_i) = \frac{\mu_0}{\pi}\sum_{m=1}^{M_1} a_m^1 D_{i,m} + \frac{\mu_0}{\pi}\sum_{m'=1}^{M'} b_{m'}\left(D^*_{i,m'} - F_{i,m'}\right) \quad (24)$$

$$B_{Z,calculated}(R_i, Z_i) = \frac{\mu_0}{\pi}\sum_{m=1}^{M_1} a_m^1 E_{i,m} + \frac{\mu_0}{\pi}\sum_{m'=1}^{M'} b_{m'}\left(E_{i,m'} - G_{i,m'}\right) \quad (25)$$

$$F_{i,m'} = \sum_{m=1}^{M_1} \alpha_{m,m'} D_{i,m} \qquad G_{i,m'} = \sum_{m=1}^{M_1} \alpha_{m,m'} E_{i,m} \quad (26)$$

with the integrals

$$D_{i,m} = \int_C \cos(m-1)\, D_R(R_i, Z_i, R_c, Z_c)\, \delta s \quad (27)$$

$$D^*_{i,m'} = \int_F \cos(m'-1)\, D_R(R_i, Z_i, R_f, Z_f)\, \delta s \quad (28)$$

$$E_{i,m} = \int_C \cos(m-1) D_z(R_i, z_i, R_c, z_c) \, ds \quad \text{- - - - -} \quad (29)$$

$$E^{*}_{i,m'} = \int_F \cos(m'-1) D_z(R_i, z_i, R_f, z_f) \, ds \quad \text{- - - -} \quad (30)$$

The calculated magnetic field depends linearly from the unknown plasma currents $b_{m'}$. We will obtain a matrix equation for $b_{m'}$ from a minimum principle so , that the deviation of the measured field in pick up coil i from the calculated value is a minimum:

$$Q = \sum_{i=1}^{I} W_i \left(b(R_i, z_i) - b_{\text{Calculated}}(R_i, z_i) \right)^2 = \text{Min} \quad \text{- - -} \quad (31)$$

$$b_{\text{calculated}} = B_{R,\text{calculated}} \cdot \cos \varphi_i + B_{z,\text{calculated}} \cdot \sin \varphi_i \quad \text{- - .} \quad (32)$$

(I = 18 is the number of pick up coils)

The orientation angle φ_i of each pick up coil is given by $\varphi_i = \Theta_i - \frac{\pi}{2}$. Θ_i is defined as angle between the normal to the vacuum vessel and the symmetry plane (see Fig. 1). The variation of Q with respect to $b_{m'}$ shall be a minimum

$$\frac{\delta Q}{\delta b_{m'}} = 0 \Rightarrow \underline{A} \cdot \vec{b} = \vec{R} \quad \text{- - - -} \quad (33)$$

This is a system of M' equations for the Fourier-components $b_{m'}$ (M is of order 4), which must be solved completely, because the vector $\vec{R}$ and the matrix $\underline{A}$ contain the measured values of b_s and the flux ψ_s , and can therefore not be precalculated.

3. Results

Calculations with the algorithm outlined in the previous chapter have been carried out for a number of test cases,where input data simulating pick up and flux-loop measurements were produced from numerical evaluation of the field of known sets of wires and external conductors, or from self-consistent equilibrium calculations.

An example of the former type is given by Fig. 2, showing flux surface contours produced by 6 top-bottom symmetric filamentary currents in the presence of the JET iron core and additional externally applied fields. The fictitious magnetic signals produced by these calculations of Christiansen /3/ using the Blum code were then fed into the present code. Although for this test case we can expect the assumption of a continuous current distribution on our control surface to lead to a maximum discrepancy, the results of our code, given in Fig. 3,give a very good fit to the "plasma surface" defined by limiter contact. Best agreement was found by restriction to Fourier modes with m up to two or three. For the reasons given in the introduction, inclusion of higher mode numbers tends to result in plasma boundaries with pronounced short wavelength oscillations.

Figs. 4 and 5 show calculations for truely self-consistent MHD equilibria. The equilibrium code used here by Zehrfeld and Casci /4/ determines resistive, steady-state MHD equilibria as a free-boundary problem, albeit presently under the assumption of a rectangular, perfect copper shell limiting the vacuum region. This feature explains the outward displacement column and the inverted D-shape of the plasma column, which make it, on the other hand,a more severe test case for our standard choice of control surface, which is optimized for centered, D-shaped JET plasmas. Again the agreement between the true and the reconstructed plasma boundary is very good (Fig. 4).

The agreement remains nearly unaltered, even when the control surface is shifted to a position strongly decentered with respect to the plasma column (Fig. 5). Also changes in size of the control surface showed an only weak influence on the plasma boundary as long as the two surfaces do not intersect.

Due to the truncation of the Fourier series, the method shows also good stability against random errors in the magnetic signals with an amplitude of 10 % of the local pick up signal or of the maximum flux difference between loops.

Clearly the algorithm outlined above cannot substitute equilibrium code versions deriving information about plasma shape and (to a limited degree) about current distributions and poloidal ß from the same measured data /5/, /6/. Requiring the toroidal plasma current density to satisfy the MHD equilibrium relation

$$j_t = R\frac{dp}{d\psi} + \frac{\mu_0}{2R}\frac{dF^2}{d\psi}$$

in terms of the two functions p and F which depend on ψ only, such MHD equilibrium codes make implicite use of information not available to the present algorithm or the so-called wire-codes. On the other hand, such full-scale free-boundary equilibrium calculations require typically two to three orders of magnitude larger computer times.

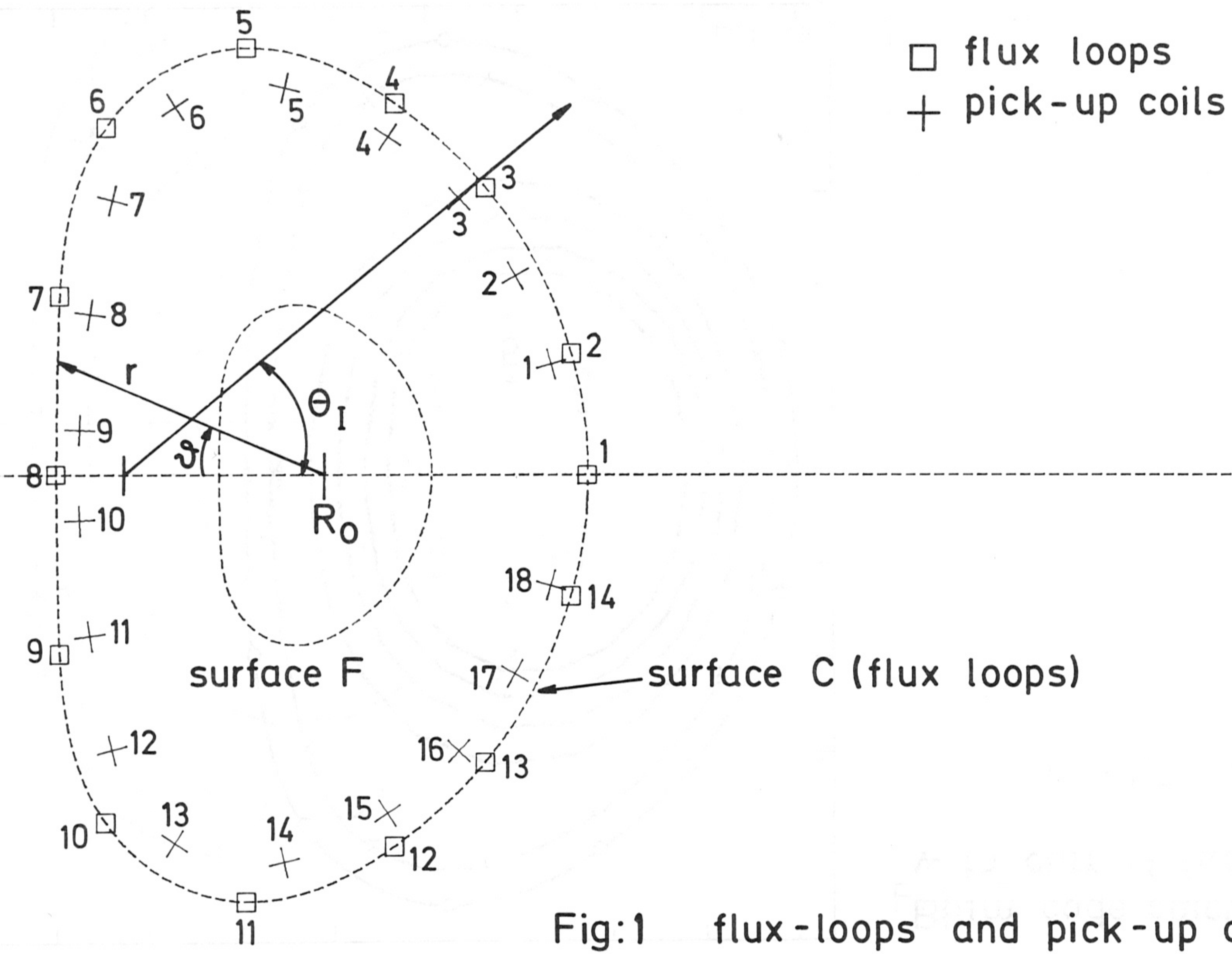
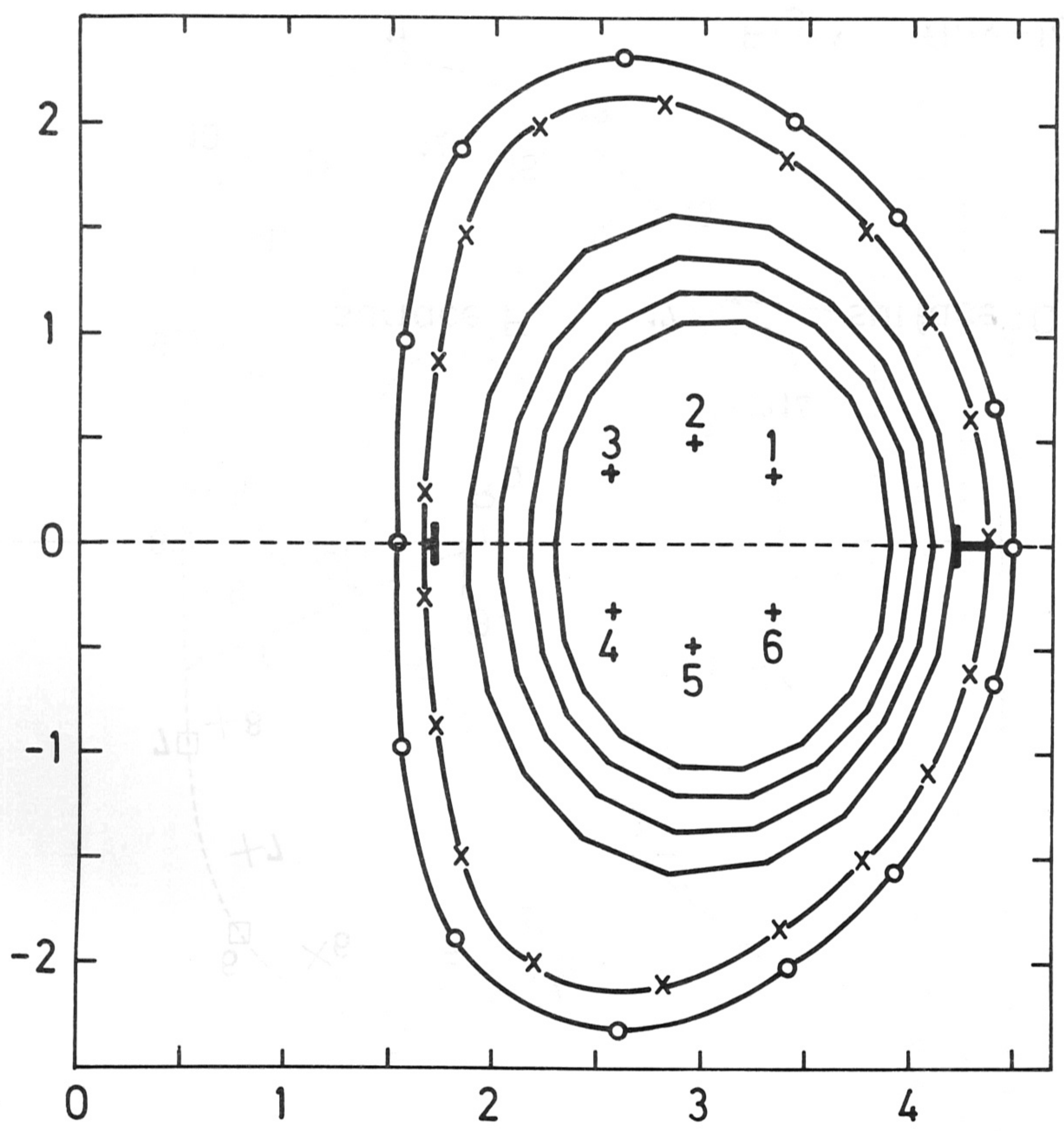


Fig:1 flux-loops and pick-up coils in JET



Blum code calculation
with current filaments

$I_t = 1{,}5$ MA

Fig: 2

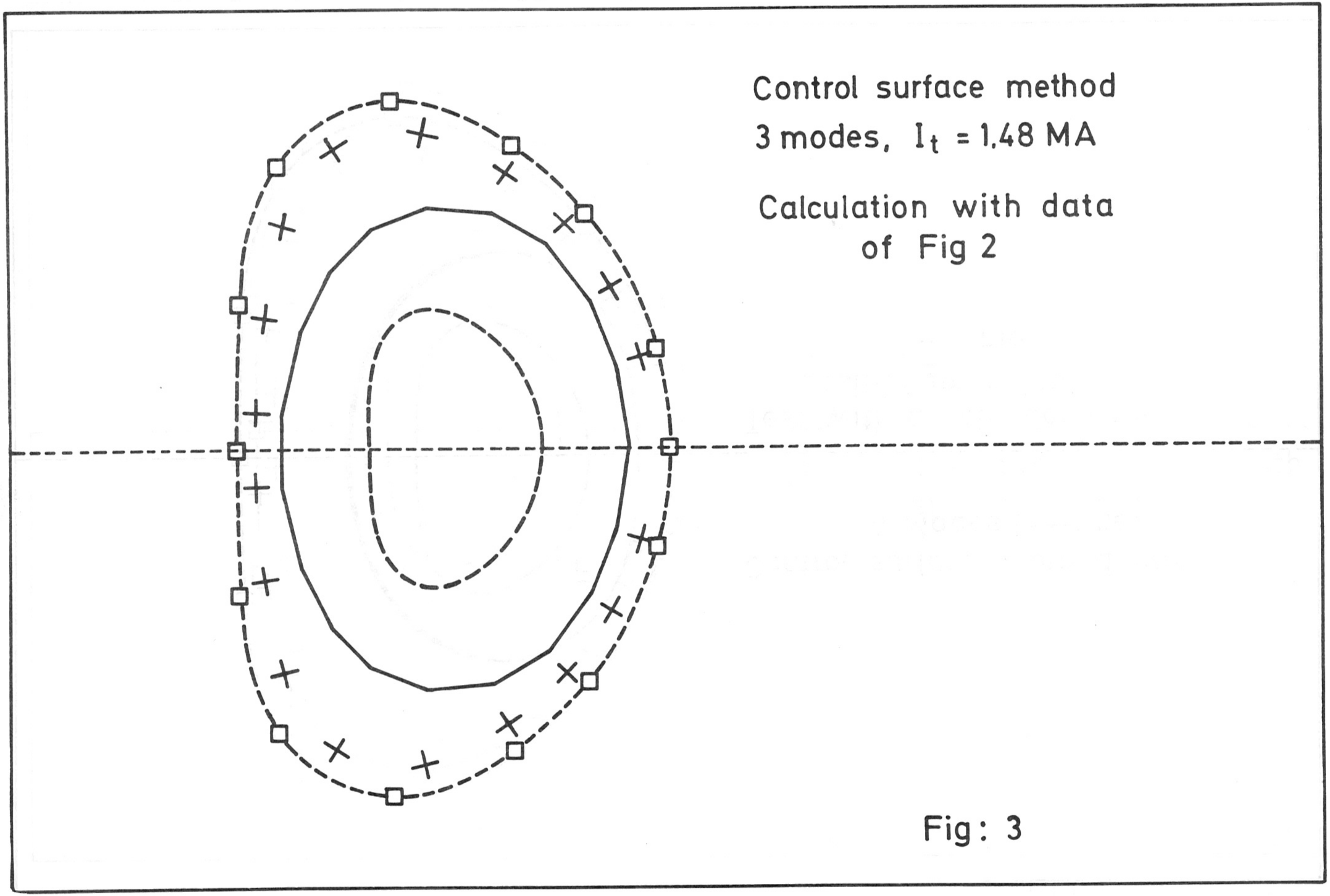

Control surface method
3 modes, I_t = 1.48 MA
Calculation with data of Fig 2

Fig: 3

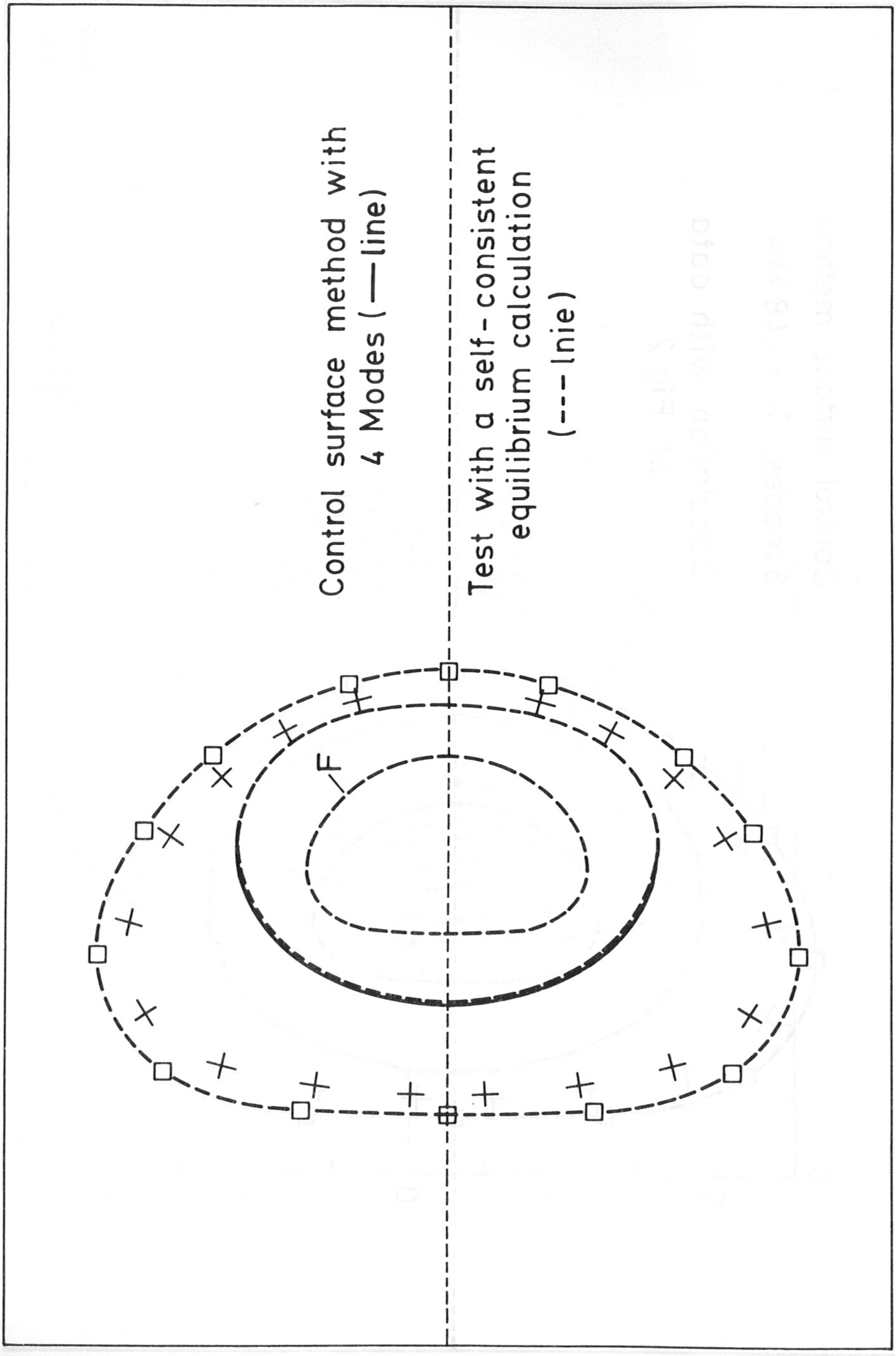

Control surface method with
4 Modes (—— line)
Test with a self - consistent
equilibrium calculation
(- - - lnie)
F

Fig: 4

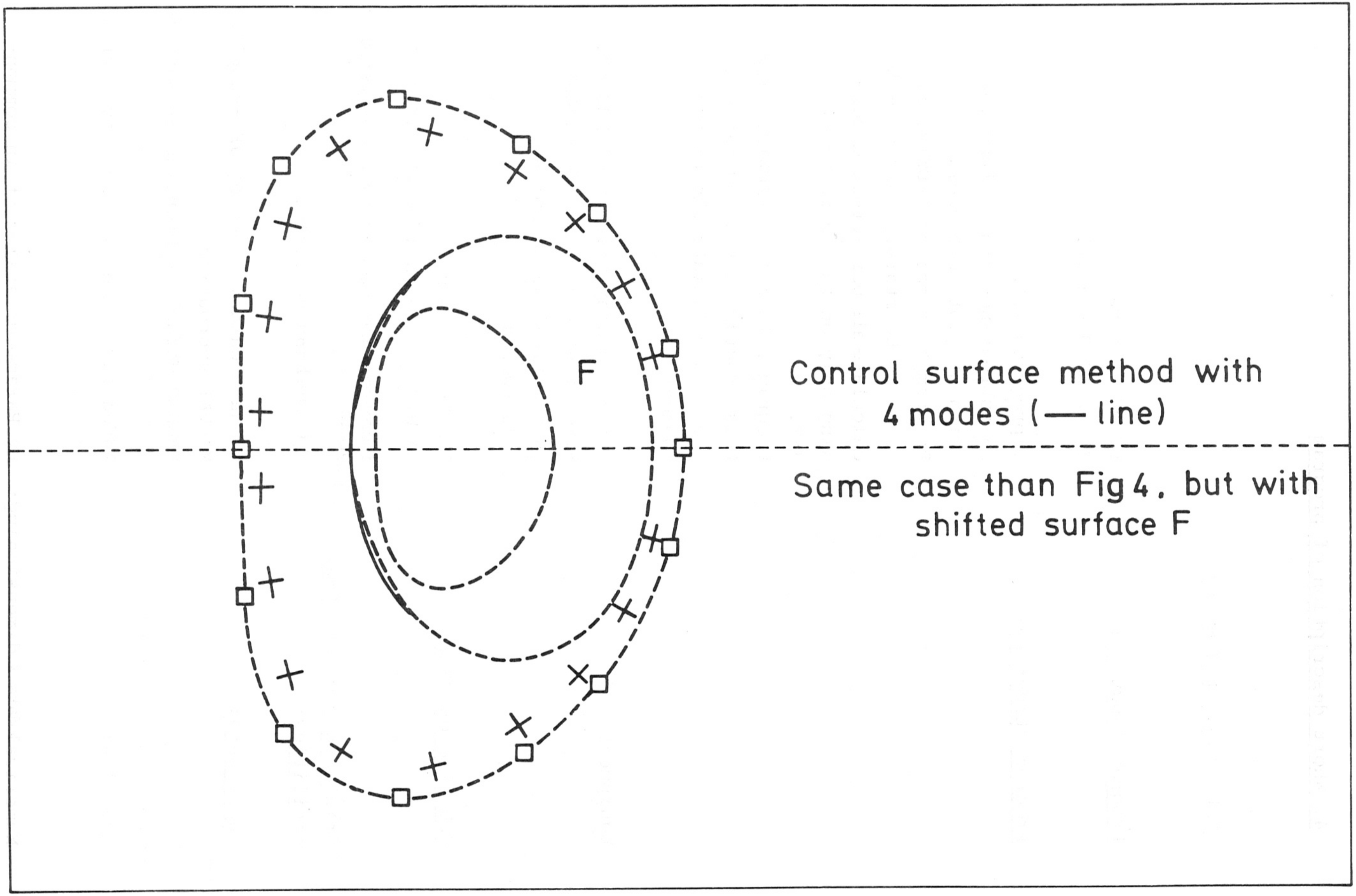

F
Control surface method with
4 modes (— line)
Same case than Fig 4, but with
shifted surface F

Fig: 5

4. Short description of program

4.1 Tabular Summary

Program language:	USANSI Fortran 77 (Ansi X3.9 - 1978)
Program structure:	2 programs with 23 subroutines and 9 functions (without plot software). Program 1 is called PREWIR. It needs to be started only once to calculate all parameters depending only on the geometry of JET. Program 2 is called LACWIR. Only this program is started between shots to calculate the plasma boundary.
Equipment:	1 disk to keep results of PREWIR. Size of storage space dependent on numerical data, normally 30 Kwords.
CPU time on CRAY:	PREWIR 3 s LACWIR 40 ms (for 200 grid points) increased time due additional diagnostic integrals
Print and plot and test facilities:	Optional and variable.
Input data:	No data cards, input by JET-supplied MAGDAT subroutine. MAGDAT default subroutine is supplied.
Examples:	Added for symmetric and asymmetric cases.
Numerical data & dimensioning:	By Parameter statement in common-file PARWIR.

4.2 Design of Program

a) Division of program into 2 steps: PREWIR and LACWIR

To meet the important requirement of short execution time, the program was divided into 2 parts.
Step 1, PREWIR program, calculates all components independently of the measured field and flux values. The precalculated matrices and vectors are permanently stored on disk.

Step 2, LACWIR program, reads those saved values and connects them with the field and flux actually measured. The plasma boundary is found and some additional test services are offered. See Fig. 6: Program and data set.

b) Symmetric and asymmetric cases

Tests show no significant gain in execution time by distinguishing between symmetric and asymmetric cases.

Most time is saved by dividing the program into 2 parts described above. Only a single program was therefore written to treat both symmetric and asymmetric cases with very short execution times. The program is also more convenient to handle if there is only one program to start between shots of JET.

c) Program parts, common blocks and names of variables

The program is of modular structure to afford good flexibility and convenient maintainance. See short description of subroutines 4.3, see also Figs. 8, 9 .

All names of subroutines and functions end with the 3 letters "WIR". Standard activities such as matrix multiplications and print of a matrix are generalized to a single each. At the beginning of a subroutine or function program purpose and input- and output data are described. Comments refer to program structure and mathematical description of chapter 2.

Data transfer between subroutines and functions is performed by labelled common blocks (except for input- and output control parameters, which are sent always by argument list).

These common blocks are activated by substitute statements.

Names of variables are valid for the whole program. See detailed description of common blocks in Section 5.5. Do loop indices are assigned to specific elements in almost all cases (I for pick-up coils, K for flux loops, N for grid points, L for plasma currents). Program labels are assigned in accordance with the program structure represented in diagrams 10 and 11.

Dimensions are variable by means of parameter-statements. See short description of common files in 4.4.

d) Test facility

For better developing and control of execution, the program was provided with a test facility that readily provides the print output of all calculations. See Section 4.5 input and output.

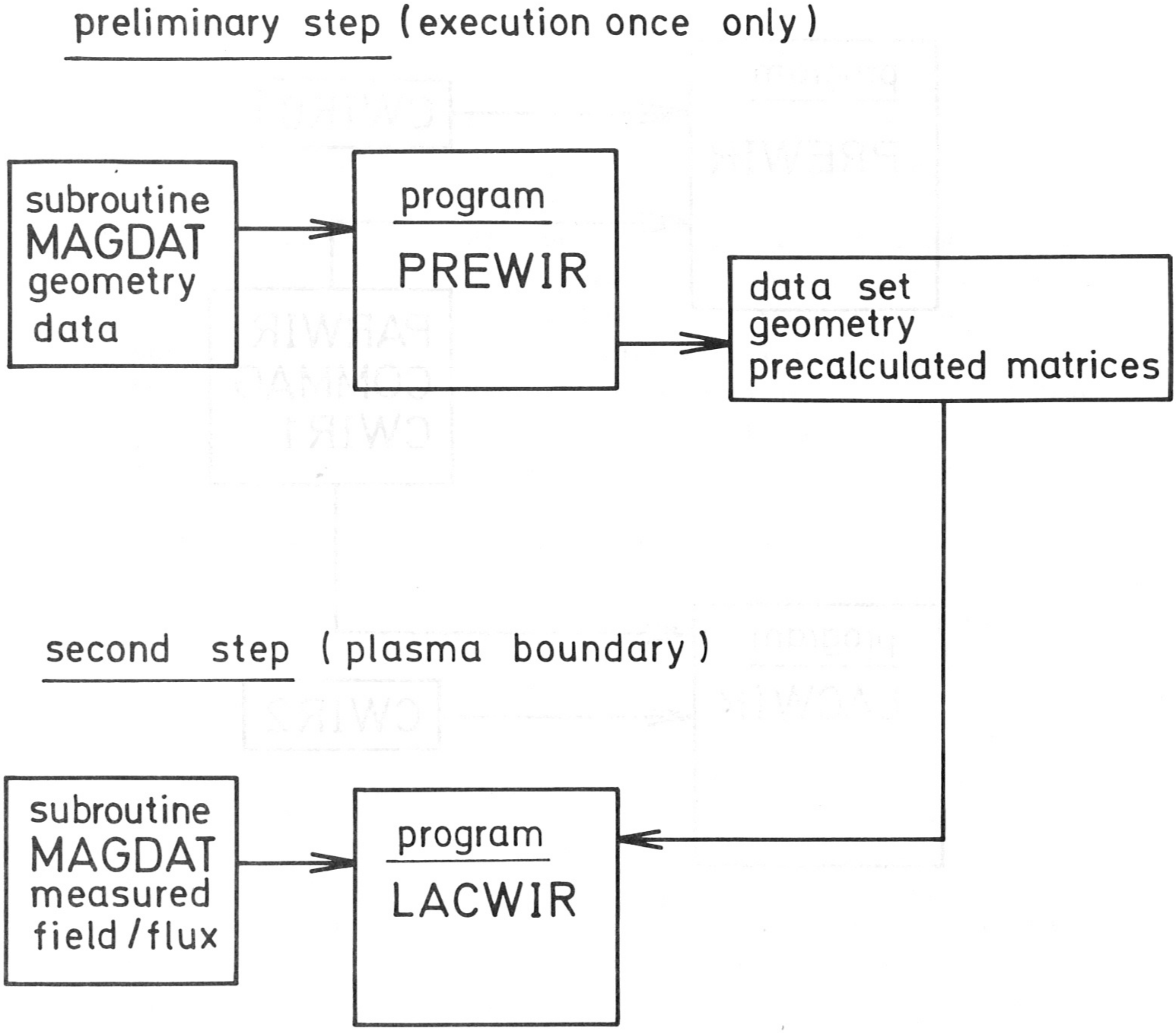


Fig: 6 program and data set

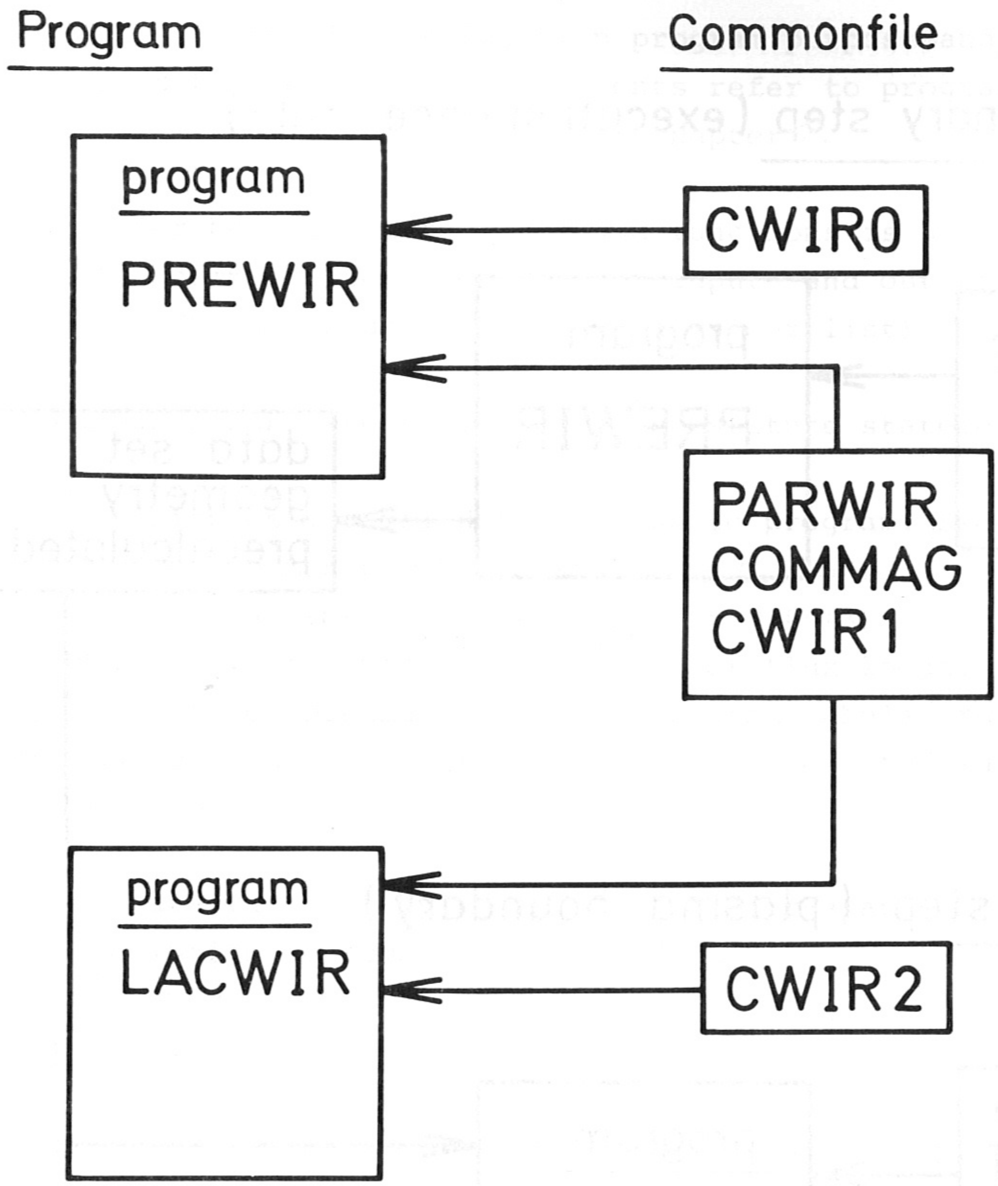


Fig: 7 program and commonfiles

4.3 Short Description of Subroutines and Functions

Program	Type	Purpose
PREWIR	SUBR	Master program to preliminary step
LACWIR	SUBR	Master program for second step to find plasma boundary
MAGDAT	SUBR	JET-supplied subroutine, defines geometry of flux loops and pick-up coils and gives measured values of field and flux
RESETR	SUBR	Fill a vector with a constant
CHEWIR	SUBR	Test if measured values of field and flux are within range allowed
GRIWIR	SUBR	Defines grid points to calculate flux in inner region
FLXWIR	SUBR	Calculates flux or derivates between flux loops - surface currents C flux loops - surface currents F pickup coils - surface currents C pickup coils - surface currents F grid points - surface currents C grid points - surface currents F
GWIR	FCT	Green function between 2 points
DGRWIR	FCT	Derivate in r of green function
DGZWIR	FCT	Derivate in z of green function
VWIR	SUBR	Elliptic integral of first and second kind
AMDWIR	FCT	Modulus K of Jacobian elliptic functions
PRTWIR	SUBR	Printout of geometry data of flux loops and pickup coils, printout of measured values and errors
PLTWIR	SUBR	Graphical representation of flux loops and pickup coils, plot of surfaces C and F.
CURWIR	SUBR	Calculation of surfaces C and F by cubic splines
TETWIR	FCT	Function value using calculated spline coefficients

BKMWIR	SUBR	Flux of surface C currents in flux loops
KMSWIR	SUBR	Flux of surface F currents in flux loops
BNMWIR	SUBR	Flux of surface C currents in grid points
NMSWIR	SUBR	Flux of surface F currents in grip points
DEWIR	SUBR	Field components Br and Bz by surface C currents
DESWIR	SUBR	Field components Br and Bz by surface F currents
NWIR	SUBR	Field components B_r and B_z by surface C currents in grid points
NSWIR	SUBR	Field components B_r and B_z by surface F currents in grid points
MTWIR	SUBR	Print a matrix with title
MXWIR	SUBR	Matrix by matrix multiplication or matrix by vector multiplication
INVWIR	SUBR	Inverse of a general matrix
CNTWIR	SUBE	Calculates a contour line in limiter point
RSMWIR	FCT	Polar coordinate r for a point in cartesian coordinates
TTKWIR	FCT	Polar coordinate ϑ for a point in cartesian coordinates

Fast Internal CRAY functions:

Equivalent subroutines and functions are supplied. The fast CRAY matrix inversion routine M x M is already exchanged by INVWIR.

SDOT	FCT	Inner product of 2 vectors
SSUM	FCT	Sum a vector

Internal IPP subroutines for plot

Dummy routines are supplied to bypass.

FRAME	New page
PLOTL	Plot a function, linear interpolation
PLOTLS	Plot a function, linear interpolation, curve is a dashed line
PLOTXT	Text on plot paper
ENDFR	Plot is finished.

4.4 Short description of common files

Data transfer between program units is executed by 5 common files: PARWIR, CWIRO, CWIR1, CWIR2, COMMAG. These files describe a set of parameter statements (PARWIR) and 4 labelled common blocks. CWIRO, CWIR1, CWIR2, COMMAG. In these all variables and vectors were finalized for the entire program. A common file is activated by a reference statement. See Figs. 7 : program and common files.

PARWIR	defines constants by parameter statements (numerical constants, constants for dimensioning). Reference to one of the labelled common blocks "COMMAG", "CWIRO", CWIR1", "CWIR2" requires reference to "PARWIR".
CWIRO	contains the labelled common block with the same name. This common block defines vectors and matrices used only in the preliminary program step.
CWIR1	this common block is the connection between the two program steps. It essentually defines precalculated vectors and matrices.
CWIR2	this common block contains vectors and matrices used only in program step 2.
COMMAG	JET-supplied common block that defines geometry and measured values for pick-up coils and flux loops. It also defines the range of values allowed.

4.5 Input and Output

a) JET has supplied MAGDAT subroutine

This subroutine defines:

- geometry data of flux loops, measured flux and instrumental error
- geometry data of pick-up coils, measured field and instrumental error.

Subroutine MAGDAT is used in both programs PREWIR and LACWIR. Defined data are transferred by labelled common block COMMAG.

Data of flux loops:

NFLMAG	- number of flux loops
TRFMAG, TZFMAG	- vectors, r- and z-coordinates
FLXMAG	- vector, measured flux
ERFMAG	- vector, instrumental error in percentage.

Data of pick-up coils:

NBPMAG	- number of pick-up coils
TRBMAG, TZBMAG	- vectors, r- and z-coordinates
BBPMAG	- vector, measured field
ERBMAG	- vector, instrumental error in percentage
Thetan	- vector, defines angle between normal to vessel and Z = 0.

b) Print and plot output control

The PREWIR and LACWIR programs contain at the beginning a data statement defining output control.

```
DATA IKEEP/9/, IPRINT /0/, IPLOT /0/, IWT /6/, ITEST /0/.
```

IKEEP - channel for data set of precalculated values

IWT - printer channel (normally 6)

IPRINT - printout control

IPRINT = 0: no printout

IPRINT ≠ 0: print of geometry, measured field/flux values, coordinates of plasma boundary and short control values.
Short control values:
- total plasma current calculated
- error of calculated field in pick-up positions
- number of modes used for minimization

IPLOT - plot output control

IPLOT = 0: no plot

IPLOT ≠ 0: plot of pick-up coils, flux loops, surfaces C and F. Additional in LACWIR: plasma boundary

ITEST - control for test printout

ITEST only works with IPRINT ≠ 0

ITEST = 0: no test print

ITEST ≠ 0: all matrices and vectors calculated are printed with title.

c) Results

The LACWIR program calculates NBPMAG+2 points (normally 20) defining the plasma boundary. The r and z components are stored in RC and ZC. The flux value is kept in H.

In addition, the r and z components of the field at all points are supplied in vectors BRC and BZC. See common CWIR2. Print output is done in accordance with the rules in the previous section. The results are preserved independent of output activities.

4.6 Program structure

a) Calling structure

The dependencies and calling structure of subroutines and functions are shown in Figs. 8 and 9.

Left side of a program name shows the calling subroutine or function. Right side shows all subroutines and functions called by this program.
For example:
GWIR is called by BKMWIR and calls VWIR and AMDWIR.

Most subroutines and functions use the common files described in 4.4 by reference statements.

b) Structure flow diagrams

The structured flow of PREWIR and LACWIR is explained in Figs. 10 and 11.

Comments and structure inside of programs refer to these pictures.
Mentioned matrices in diagrams are explained in chapter 5.4 description of common files.

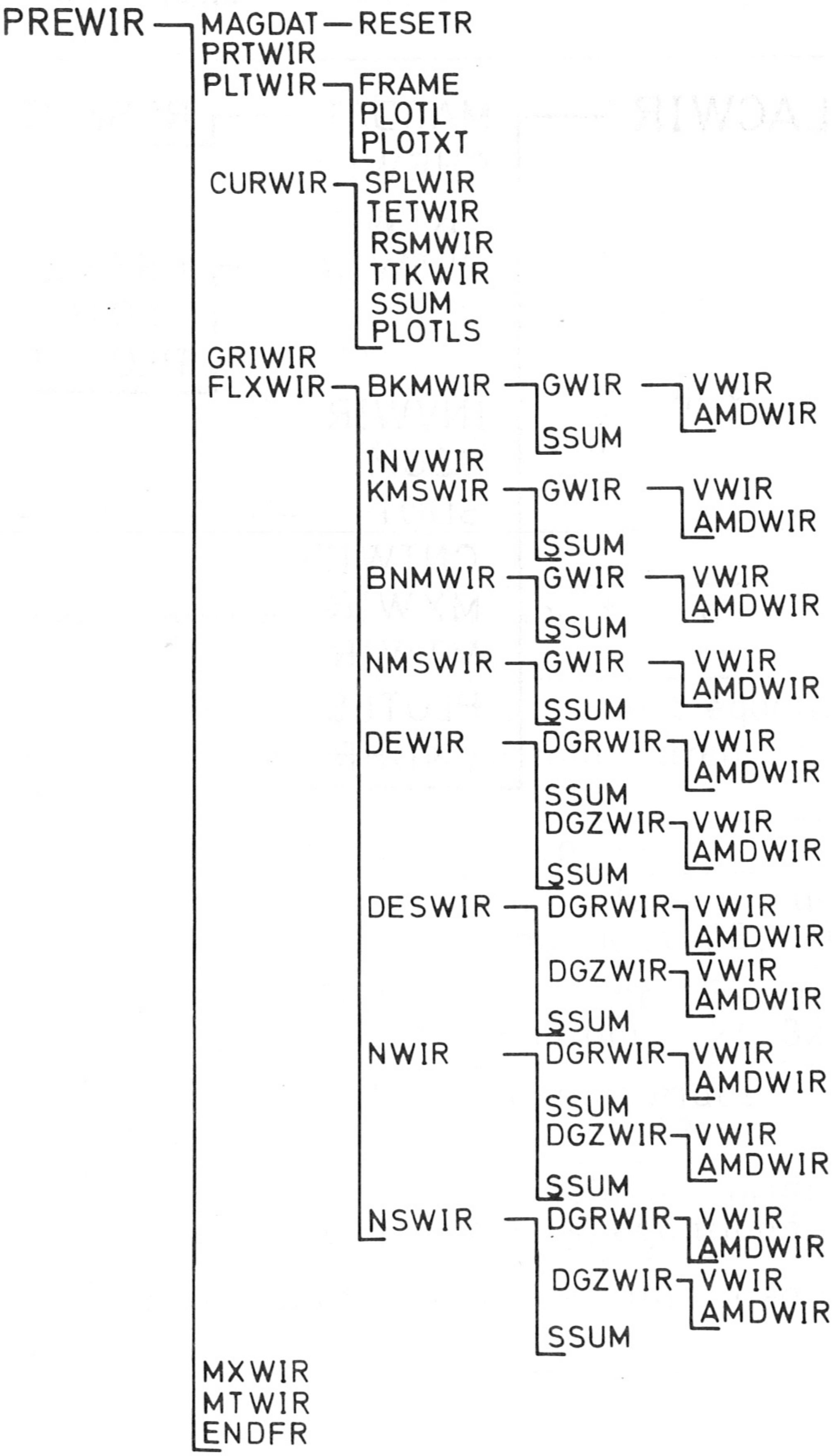


Fig: 8 structure of program PREWIR

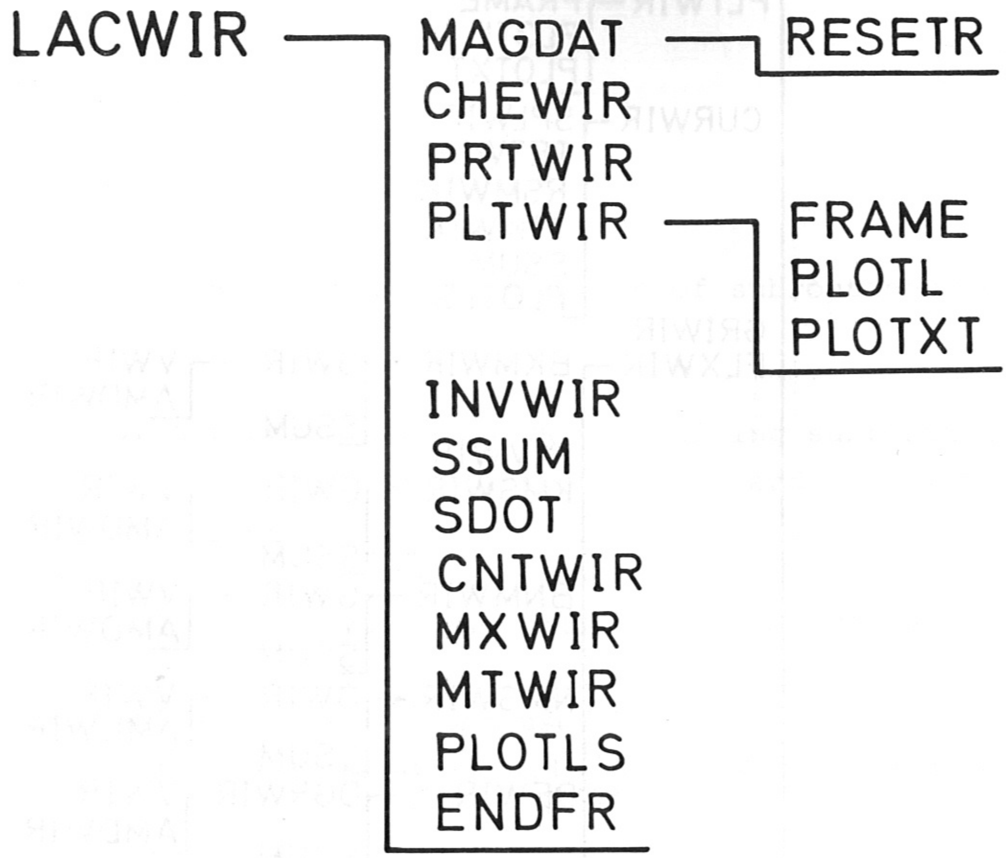


Fig:9 structure of program LACWIR

PREWIR

- get geometry data
  - JET aquisition program MAGDAT
  - standard weighting factors for instrumental error
  - angles for position of pick-up coils
  - print and plot if wished
- define surfaces C and F
- define grid points
- basis matrices for field and flux due to surface currents, equations 14, 15
- basis matrices for plasma current equations and for field and flux due to plasma current and currents outside
  - matrix ALPHA, equation 22
  - matrix TERM (field of surface C currents in pick-up positions equations 25, 32)
  - matrix FAKT (field of surface F currents in pick-up positions, equations 24, 32)
  - matrix RES (flux due to surface F currents in grid points, equations 23
  - matrices DFS, EGS (field of surface F currents in grid points, equations 24, 25)
- store precalculated matrices and geometry data

Fig: 10 structured flow diagram of PREWIR program

LACWIR

- read precalculated matrices
- get measured field and flux
  - JET - aquisition program MAGDAT
  - check measured data
  - define weighting factors (instrumental error)
  - symmetric and asymmetric values
- plot of flux loops and pick-up coils if wished
- relation flux loops and flux in inner region
- calculate plasma - currents
  - construct plasma currents equation
  - solution
  - plasma current modes
- flux in grid points
  - flux due to currents outside
  - flux due to plasma currents
  - total flux
  - print output if wished
- field in pick-up positions
  - field due to currents outside
  - field due to plasma currents
  - total field
  - print output if wished
  - compare measured field - calculated field
- field in grid points
  - field due to currents outside
  - field due to plasma currents
  - total field
  - print output if wished
- find plasma boundary

Fig:11 structured flow diagram of LACWIR program

5. Program Details

5.1 Generation of Control Surfaces

a) Polar coordinates for flux loop positions

For all flux loops positions the polar coordinates r and ϑ are calculated and stored in RK and TETK. In order to get a closed cycle, r_1 and ϑ_1 are stored additionally in RK(NFLMAG+1) and TETK (NFLMAG+1). See Fig. 12 .

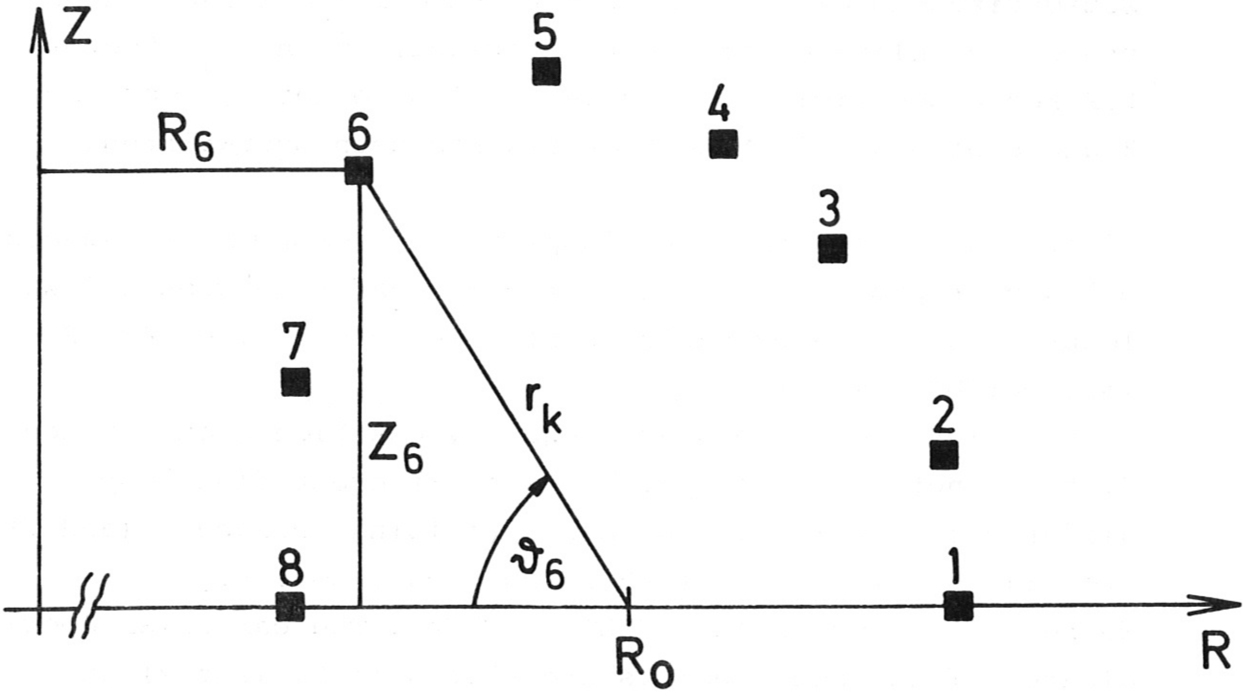


Fig. 12 : R_6, Z_6 are the coordinates of flux loop 6.

$$r_6 = (R_6 - R_0)^2 + Z_6^2$$

$$6 = \text{arctg} \frac{Z_6}{R_o - R_6}$$

b) Surfaces C and F as Polygon

For better use, reorganized values in RTK (for R) and TK (for TETK) are produced by:

RTK (K+1) = RK(NFLMAG+1-K)

TK (K+1) = TETK (NFLMAG+1-K)

K = 0,NFLMAG,1

Subroutine SPLWIR calculates for these reorganized values the cubic spline coefficients, variability in $_1$. Vector RTK keeps coefficients of order 0, B of order 1, C of order 2 and D of order 3. Note that RTK has been overwritten.

After this between the flux loops an equidistantly in spaced polygon is produced. The number of polygons between 2 flux loops is definded by variable NTET in common file PARWIR (advice: NTET = 20).
As the flux loops are nearly equally positioned on surface C, the lengths of polygons between different flux loops differ very little. The points describing surface C (and F) are placed in the middle of these polygons, note Fig. .
Number of points is IL = NTET NFLMAG. The cartesian coordinates of surface C points are stored in H1 (for x) and H2 (for z):

$$H1 = R_o - r_K \cdot \cos(\vartheta)$$
$$H2 = r_K \cdot \sin(\vartheta) .$$

To get surface F we use factor GAMMA defined in common file PARWIR.

$$P1 = R_o - r_K \cdot \cos(\vartheta) \text{ GAMMA}$$
$$P2 = r_K \cdot \sin(\vartheta) \text{ GAMMA.}$$

(r_K - polar coordinate produced by spline coefficients)

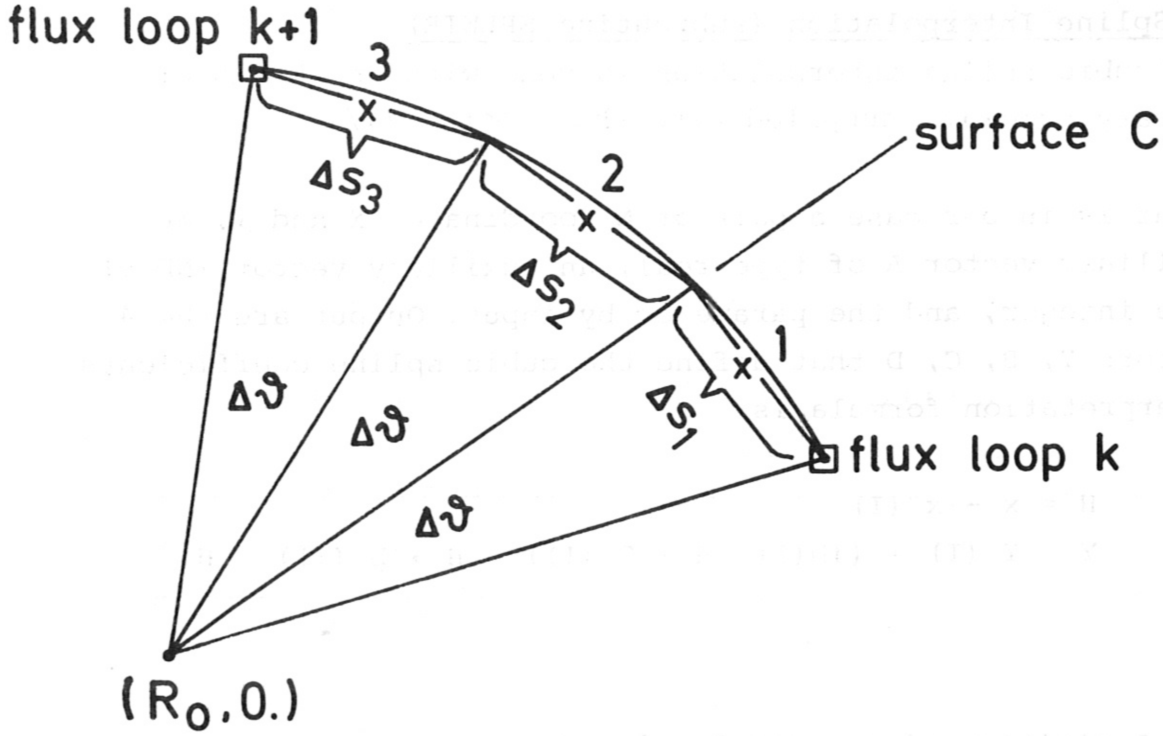


Fig: 13 approximation of surface C
(NTET = 3)

The lengths of polygons are stored to shorten execution time for integration of equations 14 and 15 (vector DELC for surface C, vector DELF for surface F).

Also the additional matrices STET and CTET facilitate integrations.

$$STET_{j\,m} = \sin(\vartheta_j \times m)$$
$$CTET_{j\,m} = \cos(\vartheta_j \times (m-1)).$$

(m = current mode, ϑ_j = polar coordinate ϑ of point j)

c) Spline Interpolation (subroutine SPLWIR)

The cubic spline interpolation is done with an IPP internal library program, supplied with the program /7/.

Input is in our case a pair of N coordinates X and Y, an auxiliary vector A of type real, an auxiliary vector IND of type integer, and the parameter by input. Output are the 4 vectors Y, B, C, D that define the cubic spline coefficients. Interpretation formula is:

H = x - x (I)

Y = Y (I) + ((D(I) H + C (I)) H + B (I)) H.

5.2 Definition of computational grid

The grid points are equidistantly spaced on rays between surface F and the centre of the pick-up coils. The distance between 2 radial points is expressed in the program by the 2 variables DX and DZ. Two additional rays are constructed to represent the z = 0-line with the limiter position at r = RLIM.

The r and z-components of the grid points are stored in the variables RG and ZG. See common CWIR1.

The numbering in radial direction starts from outside to inside. First ray is for z = 0 line, last ray connects surface F and the last pick-up coil. The total number of grid points: NN = NX * (NBPMAG+2) = NX * NZ.

Note the small shift of DX/10 and DZ/10 for the last point of each ray to avoid infinite values in surface F positions.

5.3 Tracking of plasma boundary

If we know the total flux PSITOT in all grid points, we are able to calculate the desired contour line (plasma boundary).

First the flux value H at a limiter point (RLIM,0.) is determined by linear interpolation. Then, for each ray, an index N is ascertained so that PSITOT (N) $<$ H $<$ PSITCT (N + 1), if the plasma currents is positive. Linear interpolation leads to the desired points (coordinates of plasma boundary points in RC and ZC). If there is no index N for a ray, the last point on the ray is used (point on surface F).

In addition, the r and z components of magnetic field is calculated in plasma boundary points by linear interpolation.

5.4 Evaluation of auxiliary integrals

Integration of equations 14 and 15 along surface C and F is done by trapezoidal rule of Maclaurin (order of error h^3). This is possible because the surface points are placed in the middle of a polygon spaced equidistantly in . See chapter 5.2. Integration formula:

$$\int_0^{2\pi} f\, d\vartheta \approx \sum_{k=1}^{NFLMAG} \sum_{j=1}^{NTET} f(P_j) \cdot \Delta s_j$$

Integration along surface C:

subroutine	Description	result matrices
BKMWIR	flux of surface C currents in flux loops	BKM

BNMWIR	flux of surface C currents in grid points	BNM
DEWIR	B_r and B_z components of surface C currents in pick-up coils	DI, EI
NWIR	B_r and B_z components of surface C currents in grid points	

Integration along surface F:

subroutine	description	result matrices
KMSWIR	flux of surface F currents in flux loops	BKMS
NMSWIR	flux of surface F currents in grid points	DNMS
DESWIR	B_r and B_z components of surface F currents in pick-up coils	DIS, EIS
NSWIR	B_r and B_z components of surface F currents in grid points	

5.5 Variables stored on Permanent Data Set

All data necessary to run LACWIR program are stored on a permanent data set. This includes: gemoetry data, coordinates of surfaces C and F, data of grid points, predefined field and flux data and percalculated matrices of PREWIR program.

a) Geometry data:
TRBMAG, TZBMAG, TRFMAG, TZFMAG, THETAN,
NFLMAG, NBPMAG, PHIB, SINP, COSP.

b) Coordinates of surfaces C and F and auxiliary matrices:
H1, H2, DELC, P1, P2, DELF, CTET, STET, IL.

c) Data of grid points:
RG, ZG, NX, NZ, NN.

d) Predefined field and flux data:
FLXMAG, BBPMAG, W, ERBMAG, ERFMAG, BMNMAG,
BMXMAG, FLNMAG, FMXMAG, W.

e) Precalculated Matrices:
BKM, TERM, FAKT, BNM, RES, DNM, ENM, DFS, EGS.

The meaning of these names of program variables is described in chapter 5.6: description of common files.

5.6 Description of Common Files

a) Common file CWIRO

Matrices without detailed description are of three dimensions:

dimension 1: for pick-up coils, resp. flux loops, resp. grid points

dimension 2: for surface C currents, resp. surface F currents

dimension 3: for type of mode
index = 1: cos modes
index = 2: sin modes

a) Common file CWIRO

EI - $E_{i,m}$ of equation 29 for pick-up coils

DIS - $D_{i,m'}$ of equation 28 for pick-up coils

EIS - $E_{i,m'}$ of equation 30 for pick-up coils

ALPHA - $\alpha_{m,m'}$ in equation 22
dimension 1 for surface C current modes
dimension 2 for surface F current modes
dimension 3 for type of mode

FIS - $F_{i,m'}$ of equation 26 for pick-up coils

GIS - $G_{i,m'}$ of equation 26 for pick-up coils

DNMS - $D_{i,m'}$ of equation 28 for grid points

ENMS - $E_{i,m}$, of equation 30 for grid points

FNMS - $F_{i,m}$, of equation 26 for grid points

GNMS - $G_{i,m}$, of equation 26 for grid points

B - coefficients of 1st order of cubic spline function, length: NFLMAG + 1

C - coefficients of 2nd order, length:NFLMAG + 1

D - coefficients of 3rd order, length:NFLMAG + 1

R - polar coordinates r of surface C points, length: IL = NTET * NFLMAG

TETA - polar coordinates ϑ of surface C points, length: IL

TETK - polar coordinates ϑ of flux loops, length: NFLMAG + 1

RK - polar coordinates r of flux loops, length: NFLMAG + 1

DTET - auxiliary vector to get surface C points, length: IL

RTK - reorganized polar coordinates r of flux loops, length: NFLMAG + 1

TK - reorganized polar coordinates ϑ of flux loops, length: NFLMAG + 1

HP1 - auxiliary vector for calculation of spline coefficients, length: NFLMAG + 1

IND - auxiliary integer-vector for calculation of spline coefficients, length: NFLMAG + 1

FDR - auxiliary vector for integration, stores derivates in r of Green function, length: IL

FDZ - auxiliary vector for integration, stores derivates in z of Green function, length: IL

GWF - auxiliary vector for integration, stores Green function, length: IL

VC - auxiliary vector for integration, stores subintegrals of COS modes, length: IL

VS - auxiliary vector for integration, stores subintegrals of SIN modes, length: IL

HELPO - auxiliary matrix for inversion, size: MM * MM

HELPT1 - auxiliary vector, length NBPMAG
HELPT2 - auxiliary vector, length NBPMAG

b) Common file CWIR1

BKM - $B_{n,m}$ of equation 14 for relation between surface C and flux loops

BKMS - $B_{nm'}$ of equation 15 for relation between surface F and flux loops

BNM - $B_{n,m}$ of equation 14 for relation between surface C and grid points

BNMS - $B_{n,m'}$ of equation 15 for relation between surface F and grid points

TERM - $D_{im} \cos \varsigma + E_{im} \sin \varsigma$ see equation 32

MM - number of surface C current modes
NM = (NFLMAG + 1)/2

MMS - number of surface F current modes,
NMS = NBPMAG/2

H1 - x-coordinates of surface C points,
length: IL

H2 - z-coordinates of surface C points,
length. IL

DELC - abscissa for integration, surface C points are positioned in the centre of distance DELC, length: IL

P1 - x-coordinates of surface F points,
length: IL

P2 - z-coordinates of surface F points
length: IL

DELF - analogous to DELC, for surface F,
length: IL

IL - number of surfaces C and F points,
IL = NTET *(NFLMAG + 1)

RG - x-coordinates of grid points, length NN

ZG - z-coordinates of grid points, length NN

NN - total number of grid points,
NN = NX * NZ

NX - number of grid points in radial direction

NZ - number of grid points in circumference

W - weighting factor for instrumental error of measured field, length: NBPMAG

PHIB - THETAN- /2, length NBPMAG

COSP - COS(PHIB), length NBPMAG

SINP - SIN (PHIB), length NBPMAG

THETAN - angle between normal to vessel and z = 0, at pick-up positions, length:NBPMAG

CT - auxiliary vector for plot, CT = COS (THETAN), length: NBPMAG

ST - auxiliary vector for plot, ST = SIN (THETAN), length NBPMAG

FAKT - $(D_{im'} - F_{im'}) \cos \varrho +$
$(E_{im'} - G_{im'}) \sin \varrho$
see equations 24, 32,

RES - $B_{nm'} - \frac{\mu_0}{\pi} \cdot \sum B_{nm} \alpha_{mm'}$
see equation 23,

DNM - $D_{i,m}$ of equation 27 for grid points

ENM - $E_{i,m}$ of equation 24 for grid points

DFS - (DNMS - FNMS) x ALFA 1

EGS - (ENMS - GNMS) x ALFA 1
see equation 24

LL - number of surface F current modes used
LL is fixed to 4 in program LACWIR

c) Common file CWIR2

FLXS - symmetric flux values, length; NFLMAG

FLXA - antisymmetric flux values, length: NFLMAG

BBPS - symmetric field, length: NBPMAG

BBPA - antisymmetric field, length: NBPMAG

BCAL - magnitude of total field in pick-up positions
length: NBPMAG

BCAL1 - magnitude of field in pick-up positions
due to currents

BCAL2 - magnitude of field in pick-up positions
due to plasma current, length: NBPMAG

PSITOT - total flux in grid points,
length: NN

PC - plasma current modes, solution of equation 31
dimension 1 for modes,
dimension 2 for type of mode

BC - $B^{-1}_{km}\,\Psi$ see equation 18,
dimension 1 for modes, K = 1, MM
dimension 2 for type of mode

CONST - TERM * BC, see equations 24, 25,
dimension 1 for pick-up coils
dimension 2 for type of mode

ABC - auxiliary matrix,
flux due to currents outside, later field in
grid points due to plasma current
dimension 1 for grid points, length: NN
dimension 2 for type of mode

RESPC - auxiliary matrix,
flux in grid points due to plasma current
later field in grid points due to plasma currents,
dimension 1 for grid points, length: NN
dimension 2 for type of mode

A1 - matrix for plasma current equations for cos-modes,
size LL*LL

A2 - matrix for plasma current equations for SIN-modes, size LL*LL

R1 - right hand side of plasma current equation for cos modes, length: LL

R2 - right hand side of plasma current equations for sin modes, length: LL

BR1 - field B_r in grid points due to currents outside, length: NN

BZ1 - field B_z in grid points due to currents outside, length: NN

BR - total field B_r in grid points, length: NN

BZ - total field B_z in grid points, length: NN

BRC - total field B_r in points of plasma boundary, length: NBPMAG+2

BZC - total field B_z in points of plasma boundary, length: NBPMAG+2

PCM - integrated plasma mode currents, dimension 1 for used modes, dimension 2 for type of mode

HELP3 - auxiliary vector of length NBPMAG

HELP4 - auxiliary vector of length NBPMAG

HELP5 - auxiliary vector of length NBPMAG

HELP6 - auxiliary matrix, size: LL * LL

ERR - error between measured field and calculated field, scalar

A1INV - inverse of matrix A1, size LL * LL

A2INV - inverse of matrix A2, size LL * LL

d) Common file COMMAG

BBPMAX (MAXBP) - B (poloidal) by pick-up coils

BMNMAG - permitted minimum of B (poloidal)

BMXMAG - permitted maximum of B (poloidal)

BTOMAG - toroidal vacuum B on axis R = RO, not used

ERBMAT (MAXBP) - error on B (poloidal)

ERFMAG (MAXFL) - error on flux

FLXMAG (MAXFL) - flux by loops

FMNMAG - permitted minimum of flux

FMXMAG - permitted maximum of flux

SAMMAG	-	sampling rate of magn. measurements, not used
TRBMAG (MAXBP)	-	toroidal r of pick-up coils
TRFMAG (MAXFL)	-	toroidal r of pick-up loops
TZBMAG (MAXBP)	-	toroidal z of pick-up coils
TZFMAG (MAXFL)	-	toroidal z of flux loops
VLPMAG	-	toroidal loop voltage, not used
NBPMAG	-	no. of pick-up coils
NFLMAG	-	no. of flux loops
NLMAG	-	true if data is available

c) Common file PARWIR

Constants for dimensioning of labelled common blocks

Primary:

MAXBP	-	max number of pick-up coils
MAXFL	-	max number of flux loops
MAXNN	-	max number of grid points
NTET	-	integration-points between 2 flux-loops (number of polygons for surface C between 2 flux loops)

Secondaries:

MAXMM	-	max. number of surface C current modes: MAXMM = MAXFL/2 +1
MAXMMS	-	max.number of surface F current modes: MAXMMS = MAXBP/2
MAXFLN	-	total number of surface F resp. surface C points : MAXFLN = NTET*MAXFL
Others are:		MAXFL1, MAXFL2, MAXMT, MAXM2, MAXMS2, MAXBPT

Other constants used

EPS	-	numerical constant: minimum of matrix determinant allowed
GAMMA	-	factor to construct surface F by surface C
ALFA1	-	$4 \cdot 10^{-7}$

Standard values in common file PARWIR

MAXBP = 18	MAXFL = 14
NTET = 20	MAXNN = 200
MAXMM = MAXFL/2+1	MAXMMS = MAXBP/2
MAXFLN = NTET* MAXFL	MAXFL2 = MAXFL 2
MAXM2 = 2 X MAXMM	MAXMS2 = MAXMMS 2
MAXFL1 = MAXFL+1	MAXBPT = MAXBP+2

GAMMA = 0.3

PI = 3.14159265

5.7 Formulas for elliptic integrals

Subroutine VWIR calculates the elliptic integrals of first and second kind by polynomial approximation with absolute error 2×10^{-8} /8/.

Elliptic integral of first kind is stored in program variable EK, integral of second kind in variable EE.

Subroutine ADMWIR calculates the modulus K^2 of the Jacobian elliptic integrals between 2 points (R_1, Z_1) and (R_2, Z_2) :

$$K^2 = 4 R_1 \cdot R_2 / \left[(R_1 + R_2)^3 + (Z_1 - Z_2)^2 \right] .$$

References

/1/ J. Blum, J. Le Foll, B. Thooris, Computer Physics Communications 24 (1981) 235 - 254

/2/ Stratton, Electromagnetic Theory, Mac-Graw-Hill, 1941

/3/ J.P. Christiansen, private communication

/4/ F. Casci, H.-P. Zehrfeld, this conference

/5/ H. Winter, D.B. Albert, IPP III/57, 1980

/6/ J. Blum, J. Le Foll, B. Thooris, Communication at the JET Workshop on Magnetic Field Measurements, 24th November 1980

/7/ I.N.E. Greville, Numerical Procedures for Interpolation by Spline-Functions, in Journal of the Scociety for Industrial and Applied Math., Serie B, Numerical Analysis Bd. 1, pp. 53 - 68 and j.L. Walsh, J.H. Ahlberg, E.N. Nilson, Best Approximation Properties of the Spline Fit" in Journal of Math. and Mechanics, Bd. 11, pp. 225 - 243

/7/ Handbook of Mathematical Functions, edited by Milton Abramowith and Irene Stegun, Dover Publications, Inc., New York, pp. 50 - 552