Columbia University
in the City of New York

FINITE ELEMENT COMPUTATION OF PLASMA EQUILIBRIA

MASTER

by

Monique Rivier*

1977

*(On leave from the University of Paris XI, 1974-76)

Work Supported by DOE Contract EY-76-S-02-2456

Plasma Laboratory

School of Engineering and Applied Science

Columbia University

New York, N.Y. 10027

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.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

FINITE ELEMENT COMPUTATION OF PLASMA EQUILIBRIA

Monique Rivier*

1977

NOTICE

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

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or 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.

*(On leave from the University of Paris XI, 1974-76)

Work Supported by DOE Contract EY-76-S-02-2456

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

EB

TABLE OF CONTENTS

Abstract	1
I. Introduction and formulation of the equilibrium problem	2
II. Finite element method	4
1. Weak formulation-approximate problem	7
2. Solution of the approximate problem	11
a. Choice of finite dimensional subspace	11
b. Construction of a basis	12
c. Solution of the approximate problem	13
d. Computation of the coefficients	14
III. Computation of some plasma equilibria	16
1. Without vacuum (Exponential pressure distribution)	17
2. With vacuum (Quadratic pressure distribution)	22
IV. Description of the program - Appendix	38

ABSTRACT

In this report, we investigate the applicability of the finite element method for the numerical solution of the non-linear Grad-Shafranov equation with free boundary for the flux function of a plasma at equilibrium.

This method is based on the case of variational principles and finite dimensional subspaces whose elements are piecewise polynomials functions obtained by a Lagrange type interpolation procedure over a triangulation of the domain.

Two cases of plasma pressure (exponential and quadratic including a vacuum region) were examined. In both cases we tried to show the non uniqueness of the solutions in exhibiting a deeper solution in the case of exponential pressure function, and a non-constant solution for a quadratic pressure function.

In order to get this "other" solution, we tested two linearization methods with two different constraints. Different cross sections are investigated.

1. Introduction and Formulation of the Equilibrium Problem

One of the problems posed by controlled fusion is the choice of the device used in confining the plasma. Although tokamaks seem very promising, there is not yet general agreement on the choice of the cross section; for example, many theories prove that non-circular cross sections are able to increase the critical values of the toroidal current and poloidal "beta."

The shape of the plasma section at equilibrium is one of the determining factors in the choice of device. The shape of the conducting shell, the pressure distribution, and the current distribution, for instance, influence the plasma shape which, in non-circular cross section, is not easily deduced from the shape of the conductive shell.

Methods such as the finite element method, in which the set of grid points is automatically generated from a given shape of the cross section of the shell, are useful in dealing with the first factor: the shape of the shell.

For the two other factors involved, i.e. the pressure and current functions, let us consider the model used to formulate the equilibrium problem:

Ideal magnetohydrodynamic equilibria are described by solutions of the equations

$$\begin{aligned}\vec{\nabla} p &= \vec{J} \times \vec{B} \\ \vec{\nabla} \times \vec{B} &= \mu_0 \vec{J} \\ \vec{\nabla} \cdot \vec{B} &= 0\end{aligned}$$

where $P(\vec{x})$ is the pressure function, $B(\vec{x})$ the magnetic field and $J(\vec{x})$ the electric current density.

Assuming an axial symmetry for the system and using a cylindrical coordinate system (r, z) , the preceding system becomes the "Grad Shafranov" equation which describes the space variation of the poloidal flux function ψ (after normalization) [4]

$$(1) \quad L(\psi) = r \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \psi}{\partial r} \right) + \frac{\partial^2 \psi}{\partial z^2} = - \beta r^2 \frac{dP(\psi)}{d\psi} - \gamma \chi \frac{d\chi(\psi)}{d\psi}$$

Here P is the plasma pressure and χ the poloidal current (or toroidal field). The magnetic field is defined in terms of χ and ψ by

$$\vec{B} = \frac{1}{r} \chi \vec{e}_\phi - \frac{1}{r} (\vec{e}_\phi \times \nabla \psi)$$

At equilibrium χ and P are well known to be functions of ψ only.

The fundamental question is then: given a pressure distribution and a current distribution, what is the shape of the plasma cross section for a given boundary condition?

In most of the cases, this question has no satisfactory answer:

First, the equation involved is an elliptic nonlinear equation: from this nonlinearity arises the problem of non uniqueness of the solutions. So there can be multiple shapes

of plasma cross sections. In addition, the number of solutions, where it can be proved that there is more than 1, cannot be predicted from a general mathematical theory. This phenomena called bifurcation, is the possibility of several distinct solutions from one set of boundary values. The objectives of this work were to introduce in this laboratory a finite element program, and to obtain from it some plasma equilibria where the bifurcation phenomena occurs, such computations having already been done for most of them by the finite difference method.[8]

In Section II we will develop the finite element method and its application to the Grad Shafranov Equation.

In Section III results of computation are presented for plasma equilibria in shells of various shapes and some bifurcated solutions are exhibited. Comparisons are made with solutions obtained by finite differences.

Section IV is the description of the computer program.

II. Finite Element Method [1]

The traditional approach to the numerical solutions of equation of form (1) has been through the use of finite difference methods.

Usually the differential operator L is replaced by a five-point difference approximation. The truncation error which occurs in this method is 2nd order in the mesh size.

The methods giving a smaller truncation error, i.e. error proportional to a higher power of the mesh size, generally use a larger number of points; for example, the 9-points difference approximation for 4th order accuracy. But this method involves a discretized operator which is not self-adjoint, though the initial differential operator is self-adjoint, and so we might lose some convergence properties of the solution; in addition, inverting the 9 point matrix is not so easy.

Another way to preserve the character of the operator and to have a high order approximation is the Ritz-Galerkin method, and, more precisely, the finite element method.

The first step of the procedure is to reformulate the problem into the so-called weak form, and then look for a weak solution in some appropriate Hilbert space. This reformulation of the problem is equivalent to a variational formulation of the problem in the case of a self-adjoint differential operator.

This reformulation weakens the differentiability and continuity properties imposed on the solution, and so it makes the second step easier, which consists of a choice of a particular finite dimensional subspace of the Hilbert space, in which an approximate solution of the boundary problem is subsequently sought.

In the construction of the finite dimensional subspace, we will use finite elements; in this report we have restricted ourselves to finite elements of order 1, i.e., piecewise linear functions. The main advantage of the finite element method is that the grid points are not given arbitrarily but are chosen to conform to the shape of the domain itself and so they fit

the boundary much better than do the rectangular or polar grid points usually used in finite difference methods.

On the other hand, if the support of the basis functions are hexagons as in the case of a triangular grid, the matrix will have at least 7 non zero terms on each line. So this matrix is not so easy to invert. But it should be noted that in order to increase the accuracy, we often don't need to increase the number of points. Thus in order to increase the degree of the Lagrange interpolation polynomials used, the dimension of the matrix does not increase, but the number of non zero elements of the matrix increases.

Because of this difficulty in inversion, the finite element method is still only rarely used for time dependent problems. The Galerkin method in this case of time dependent problems produces a system of ordinary differential equation for the time dependent coefficients $\xi_i(t)$ which has to be solved also by an approximation method.

1. Weak formulation and approximate problem

First of all let us simplify equation (1) by means of a change of variable such as: [5]

$$x = \frac{r^2}{2} \quad y = \sqrt{2} z$$

the equation

$$r \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \psi}{\partial r} \right) + \frac{\partial^2 \psi}{\partial z^2} = -\beta r^2 \frac{dP}{d\psi} - \gamma x \frac{\partial x}{\partial \psi}$$

becomes

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{1}{x} \frac{\partial^2 \psi}{\partial y^2} = -\beta \frac{dP}{d\psi} - \gamma x \frac{\partial x}{\partial \psi}$$

We restrict ourselves to the simple case of $\psi = G$ on the boundary $\partial\Omega$ of the domain Ω . Then setting $\psi - G$ as the new unknown, we get a homogenous Dirichlet boundary condition problem.

$$\begin{aligned} (2) \quad & \left[\begin{aligned} - \left(\frac{\partial^2 \psi}{\partial x^2} + \frac{1}{x} \frac{\partial^2 \psi}{\partial y^2} \right) &= \beta \frac{dP}{d\psi} (\psi + G) + \gamma \frac{dx^2}{2d\psi} (\psi + G) = f(\psi, G) \\ (3) \quad & \psi = 0 \quad \text{on } \partial\Omega \end{aligned} \right. \end{aligned}$$

We will give the weak formulation of the problem in precise form. Let $L^2(\Omega)$ be the Hilbert space of functions which are square integrable over Ω . The inner product on $L^2(\Omega)$ is given by

$$(u, v)_{L^2(\Omega)} = \int_{\Omega} u(x, y) v(x, y) \, dx \, dy$$

We define $D(L)$ as the set of functions in $L^2(\Omega)$ which have the following properties:

- 1) they are twice continuously differentiable in Ω
- 2) the functions and their first derivatives are continuous on $\partial\Omega$
- 3) they satisfy the boundary condition (3)

The linear differential operator L is defined as

$$L\psi(x, y) = -\left(\frac{\partial^2 \psi}{\partial x^2} + \frac{1}{x} \frac{\partial^2 \psi}{\partial y^2}\right)$$

on the domain $D(L)$

Then for the classical solution, we have the following,

Problem:

To find the function $\psi \in D(L)$ which satisfies the equation

$$L\psi(x, y) = f(\psi, G).$$

Let us recall some definitions from functional analysis. (Ref. [2])

Definition: $H_0^1(\Omega)$ denotes the Hilbert space of all elements of $L^2(\Omega)$ that have generalized derivatives of the first order in $L^2(\Omega)$ and which vanish on the boundary $\partial\Omega$. The inner product in $H_0^1(\Omega)$ is defined by

$$\langle u, v \rangle = \int_{\Omega} (uv + \nabla u \cdot \nabla v) \, dx \, dy$$

Now let us define the following integral identity

$$(4) \quad \int_{\Omega} \left(\frac{\partial \psi}{\partial x} \frac{\partial \varphi}{\partial x} + \frac{1}{x} \frac{\partial \psi}{\partial y} \frac{\partial \varphi}{\partial y} \right) dx dy = \int_{\Omega} f(\psi, G) \varphi dx dy$$

$$\forall \varphi \in D(L)$$

If the equation (2) was linear, i.e. f independent of ψ , it is easy to see that any ψ satisfying (4) is a solution to the problem (2) (3). However the integral occurring in the left hand side is defined for an arbitrary function $\psi \in H_0^1(\Omega)$. Therefore, a natural class of generalized solutions of Eq. (4) is the class of generalized solutions in $H_0^1(\Omega) \cap V$ where V is the class of functions for which $\int_{\Omega} f(\psi, G) \varphi dx dy$ is well defined. For ψ and $\varphi \in H_0^1(\Omega) \cap V$ we define the bilinear form

$$a(\psi, \varphi) = \int_{\Omega} \left(\frac{\partial \psi}{\partial x} \frac{\partial \varphi}{\partial x} + \frac{1}{x} \frac{\partial \psi}{\partial y} \frac{\partial \varphi}{\partial y} \right) dx dy$$

Then we have the following

Generalized Problem:

To find function $\psi \in H_0^1(\Omega) \cap V$ which satisfies the equation

$$a(\psi, \varphi) = \int_{\Omega} f(\psi, G) \varphi(x, y) dx dy \quad \forall \varphi \in H_0^1(\Omega) \cap V$$

Note that this bilinear form is positive definite, symmetric and continuous.

We shall now define the Galerkin type approximation procedure. Let M_N be any finite dimensional subspace of $H_0^1(\Omega) \cap V$ then we have the following.

Approximate problem: To find $\psi \in M_N$ which satisfies the equation

$$a(\psi, \varphi) = \int_{\Omega} f(\psi, G) \varphi \, dx \, dy \quad \forall \varphi \in M_N$$

2. Solution of the approximate problem

a. Choice of a finite dimensional subspace

The next step consists of the choice of the finite dimensional subspace M_N and the construction of a basis for it. Let us choose subspaces generated by piecewise polynomials functions of a specified degree over the domain Ω . We consider first a finite partition of Ω into polygonal subdomains of elementary shape (triangles, rectangles) and then ψ is equal to a polynomial function of specified degree in each polygonal subdomain of the partition.

When each polynomial is uniquely determined by its behavior within its associated polygon, this approximation is commonly referred to as the finite element method.

The most common choice of subdomain shapes are triangles and rectangles. In order to preserve the capability of solving the equation domains, of arbitrary boundary curves, we will consider the case of a triangular partition of Ω .

Then the finite dimensional subspace will consist of functions ψ which are piecewise polynomials on $\bar{\Omega}$ and which are polynomials of a specified degree as on each triangle of the triangulation of Ω . On each triangle ψ is completely and uniquely determined by its local behavior that is, by its behavior within the triangle.

b. Construction of a basis

Next we construct a basis for the finite dimensional subspace M_N . Suppose the domain Ω has been partitioned into triangles. Let m be the desired degree of the piecewise polynomial functions in M .

In order to uniquely determine $\psi \in M_N$ from its local behavior, we need in each triangle $\nu = \frac{1}{2}(m+1)(m+2)$ values, as a polynomial of m degree has exactly ν degrees of freedom. These data can be taken as the values of the function or of its derivatives at a certain number of points in the triangle.

We will use only function values and no derivative values in order to determine the polynomials. This is called a Lagrange type interpolation. In fact, in this report we shall use polynomials of degree $m=1$. So $\nu=3$, and we choose values at the three vertices of the triangles. For future refinement $m > 1$ may be used.

To each node $I=1, N$ we will now associate a basis function $w_I(x,y)$, which has minimal support, such that w_I vanishes outside the union of the triangles to which the I node belongs. w_I is continuous, with derivatives piecewise continuous, has the value 1 at the I^{th} node and 0 at the other neighbour nodes and is linear in each triangle possessing the I^{th} node as a vertex.

Denote by σ_l the support of w_l : $\sigma_l = \bigcup_{k \in K} T_l^k$, T_l^k being the triangles (total no. K_l) to which the l^{th} node belongs. w_l has a different expression on each triangle composing its support. In each triangle T_l^k , w_l is a polynomial of degree 1 and so has the general form

$$(7) \quad w_l(x, y) = \alpha_l^k x + \beta_l^k y + \gamma_l^k$$

Let J and K be the other two vertices of the triangle T_l^k . and let x_J, y_J, x_K, y_K and x_l, y_l be the coordinates of the nodes J, K and l . Then the coefficients $\alpha_l^k, \beta_l^k, \gamma_l^k$ will be determined by the system

$$(8) \quad \alpha_l^k x_l + \beta_l^k y_l + \gamma_l^k = 1$$

$$(9) \quad \alpha_l^k x_J + \beta_l^k y_J + \gamma_l^k = 0$$

$$(10) \quad \alpha_l^k x_K + \beta_l^k y_K + \gamma_l^k = 0$$

c. Solution of the approximate problem

The solution ψ of the approximate problem (6) will be, as in any Galerkin approximation, of the form

$$(11) \quad \psi(x, y) = \sum_{l=1}^N \xi_l w_l(x, y)$$

where $\{w_l\}_{l=1}^N$ is the basis of the space M_N .

As $w_i(x_i, y_i) = 1$, the coefficient ξ_i is in fact equal to $\psi(i)$ i.e. the value of ψ at each node of the triangulation.

Substituting the expansion (11) into Eq. (6) and taking for the test functions successively each of the basis function w_j of M_N , we obtain a system of equations for computing the $\psi(i)$ or ξ_i .

$$(12) \quad \sum_{i=1}^N \xi_i a(w_i, w_j) = \int_{\Omega} f\left(\sum_{i=1}^N \xi_i w_i, G\right) w_j \, dx \, dy \quad \forall j = 1, N$$

As this system of equations is nonlinear, we use a method of linearization in order to find the coefficients ξ_i . The flux function $\psi(x, y)$ will then be computed by the expansion form (11).

d. Computation of the coefficients ξ_i

We begin by observing that the bilinear form $a(w_i, w_j)$ is nonzero only if the supports of w_i and w_j have at least one triangle in common. L_{ij} the set of indices L of those triangles which compose the intersection of the domain of w_i and w_j .

Denote by L_{ij} the set of indices L of those triangles T^L which compose the intersection of the supports of w_i and w_j

$$\bigcup_{L \in L_{ij}} T^L = \sigma_i \cap \sigma_j$$

$$a(w_I, w_J) = \sum_{\ell \in L_{IJ}} \int_{T^\ell} \left(\frac{\partial w_I}{\partial x}(x, y) \frac{\partial w_J}{\partial x}(x, y) + \frac{1}{x} \frac{\partial w_I}{\partial y}(x, y) \frac{\partial w_J}{\partial y}(x, y) \right) dx dy$$

$$\text{On } T^\ell \quad w_I(x, y) = \alpha_I^\ell x + \beta_I^\ell y + \gamma_I^\ell$$

$$w_J(x, y) = \alpha_J^\ell x + \beta_J^\ell y + \gamma_J^\ell$$

$$\text{So} \quad a(w_I, w_J) = \sum_{\ell \in L_{IJ}} \int_{T^\ell} \left(\alpha_I^\ell \alpha_J^\ell + \frac{1}{x} \beta_I^\ell \beta_J^\ell \right) dx dy$$

The right hand side $\int_{\Omega} f(\psi, G) w_J(x, y) dx dy$ is in fact an integration not over Ω but over the support of w_J : $\sigma_J = \bigcup_{k \in K_J} T_J^k$. On each T_J^k composing this support ψ is a linear combination of the values of ψ on each vertex.

On each triangle T_k^J with vertices JA, JB, JC

$$\psi(x, y) = \psi(JA) w_{JA}(x, y) + \psi(JB) w_{JB}(x, y) + \psi(JC) w_{JC}(x, y)$$

$$\text{So} \quad \int_{\Omega} f\left(\sum_{l=1}^N \xi_l w_l, G\right) w_J dx dy = \sum_{k \in K_J} \int_{T_k^J} f(\xi_{IA} w_{IA} + \xi_{IB} w_{IB} + \xi_{IC} w_{IC}, G) w_J dx dy$$

(13)

Now in order to compute this right hand side let us go back to the physical meaning of f .

III. Computations of some Plasma Equilibria

In the Grad Shafranov equation (1)

$$L\psi = -\beta r^2 \frac{dP(\psi)}{d\psi} - \gamma x \frac{dx}{d\psi}$$

β and γ are constant

$$\psi = G \text{ on the boundary } \partial\Omega.$$

The choice of the pressure and current distributions could be taken arbitrary in an equilibrium study, but in fact distributions are determined by the initial configuration and the time evolution of the plasma.

In the following study, we have restricted ourselves to two cases of pressure function, both avoiding sharp boundary profiles but using smooth functions which remain positive.

A sufficient condition prohibiting a maximum of ψ in the interior of the domain of solution, i.e. for ψ to attain its maximum on the boundary $\partial\Omega$, is that the right hand side of equation (6) be positive definite. (Ref [3]) In both cases, our right hand side will satisfy this condition when $P(\psi)$ and $\chi^2(\psi)$ are monotone decreasing functions.

As we are excluding solutions with saddle points, by the choice of our $P(\psi)$ and $\chi^2(\psi)$ solution will be valley shaped with a single minimum inside the domain and will attain their maximum on the boundary $\partial\Omega$.

1. Exponential pressure profile without vacuum

In equation (7), we choose as an example $P(\psi)$ to be of the form $e^{-\psi}$ and we choose zero poloidal current, or $\gamma = 0$. Thus we get

$$(14) \quad \begin{cases} L\psi = + \beta r^2 e^{-\psi} \\ \psi = G \text{ on } \partial\Omega \end{cases}$$

After our change of variables and function (p7), the equation (14) becomes

$$(15) \quad \begin{cases} -\left(\frac{\delta^2 \psi}{\delta x^2} + \frac{1}{x} \frac{\delta^2 \psi}{\delta y^2}\right) = -\beta e^{-(\psi + G)} \\ \psi = 0 \text{ on } \delta\Omega \end{cases}$$

The weak formulation is therefore:

Find $\psi(x, y) = \sum_{l=1}^N \xi_l w_l(x, y)$ such as

$$\left[\begin{array}{l} a(\psi, w_J) = -\beta e^{-G} \int_{\Omega} e^{-\psi} w_J dx dy \quad \forall J = 1, N \\ \psi = 0 \text{ on } \delta\Omega \\ \sum_{i=1}^N \xi_i \int_{\Omega} \left(\frac{\delta w_i}{\delta x} \frac{\delta w_J}{\delta x} + \frac{1}{x} \frac{\delta w_i}{\delta y} \frac{\delta w_J}{\delta y} \right) dx dy = -\beta e^{-G} \int_{\Omega} e^{-\sum \xi_i w_i} w_J dx dy \\ \forall J = 1, N \\ \psi = 0 \text{ on } \delta\Omega \end{array} \right.$$

Marder and Weitzner [5], Suzuki [9], Reusch [8], studied the equation (1) and proposed numerical methods for computing bifurcated solutions, as a standard numerical technique gives only one solution, usually the shallowest one. (Ref. [10])

As a test of the finite element method, we calculate examples already computed in this laboratory by finite difference method [8]. So the constraint imposed, in order to get each of the solutions, was the same as in [8], that is to say the minimum value of ψ at each iteration.

During iteration the minimum of ψ was kept fixed in value, but not in location while the value on the boundary G was allowed to float.

The linear system (12) obtained after the linearization process is solved by an over relaxation method with the calculation of an optimal relaxation parameter.

(If G is simply held constant during iteration, the iteration generally converges to the shallowest solution).

So the linearization method applied to the formulation (16) is: (G remains constant here)

ψ^n supposed known, we set

$$\psi^{n+\frac{1}{2}} = \psi^n - \min \psi^n + \min \psi^{n-1}$$

$$\text{So } \min \psi^{n+\frac{1}{2}} = \min \psi^{n-1}$$

and then

$$\left[\begin{array}{l} a(\psi^{n+1}, w_J) = -\beta e^{-G^{n+1}} \int_{\Omega} e^{-\psi^{n+1/2}} w_J dx dy \\ \psi^{n+1} = 0 \text{ on } \delta\Omega \end{array} \right.$$

Using the definition of $\psi^{n+1/2}$, we get

$$\begin{aligned} a(\psi^{n+1}, w_J) &= -\beta e^{-G^n - 1} \int_{\Omega} e^{-\psi^n - \min \psi^n + \min \psi^{n-1}} w_J dx dy \\ &= -\beta e^{-(G^n - 1 + \min \psi^n - \min \psi^{n-1})} \int_{\Omega} e^{-\psi^n} w_J dx dy \\ &= -\beta e^{-G^n} \int_{\Omega} e^{-\psi^n} w_J dx dy \\ \psi^{n+1} &= 0 \text{ on } \delta\Omega \end{aligned}$$

And so the iteration method is described by

(17)	$G^n = G^{n-1} - \min \psi^n + \min \psi^{n-1}$
(18)	$a(\psi^{n+1}, w_J) = -\beta e^{-G^n} \int_{\Omega} e^{-\psi^n} w_J dx dy$
(19)	$\psi^{n+1} = 0 \text{ on } \Omega$

Using this method, we get $\min \psi^{n+1} + G^{n+1} = \min \psi^0 + G^0$

The initial guess (ψ^0, G^0) has been chosen as a single pyramid over the domain Ω i.e.

$$\psi^0(x, y) = 5 w_1(x, y)$$

ξ is the minimum of ψ . ($\xi = \psi^0(1)$). In this special case of pressure function, we get an explicit representation of G^0 function of the minimum of ψ^0 : ξ

$$a(\psi^0, w_1) = -\beta e^{G^0} \int_{\Omega} e^{-\xi w_1} w_1 dx dy$$

$$a(\psi^0, w_1) = \xi a(w_1, w_1)$$

$$G^0 = -\log \frac{\xi a(w_1, w_1)}{-\beta \left(\int_{\Omega} e^{-\xi w_1} w_1 dx dy \right)}$$

We did not put any symmetry constraint on the solution but we get in every case a symmetric solution in z .

Figure (1) shows the results obtained by finite elements with 64 triangles and compared with the finite difference method. The domain is rectangular

and the parameter β was chosen as 0.5.

The curves with little squares was obtained by finite difference and the one with little triangles by finite elements. The initial values were $\psi^0 = \xi w_1(x, y)$

The figure is now to be used in the other way. Given a boundary value larger than -8, for instance $G = -0.7$, then we get two different solutions, one with a minimum equal to 2.8 and another one with a minimum equal to 1.6.

As we can see on figure (1), if the boundary value G is chosen lower than the critical value, (the only value for which there is only one solution), there is no solution at all. This behavior is particular to the $e^{-\psi}$ form of the function $f(\psi)$.

Depending on the size of the domain, the problem with this form of f and homogeneous boundary data has either no solution or else possesses a finite number of solutions [3]. The solution is never unique except for the critical value of G .

2. Quadratic pressure profile with vacuum:

We consider another example where $P(\psi)$ and $\chi(\psi)$ are quadratic functions of ψ with coefficients independent of r and where the vacuum magnetic field containing the plasma is surrounded by a continuous ideally conductive shell.

Without loss of generality, we can choose the plasma surface, where $P = 0$ by the surface of $\psi = 0$. With

$$P(\psi) = \alpha \psi^2 \text{ and } \chi(\psi) = \alpha_1 \psi^2 + \beta$$

the equation (1) becomes

$$L(\psi) = -\lambda(r^2 - K) \psi$$

In the vacuum region surrounding the plasma, the flux function must satisfy:

$$L\psi = 0.$$

With the boundary condition on the shell surface $\psi = G$ and with the boundary conditions on the plasma surface that $\psi = 0$ and the first derivatives of ψ are continuous. So we can pose the problem as:

$$(20) \quad L\psi = \begin{cases} -\lambda(r^2 - K) \psi & \text{if } \psi \leq 0 \\ 0 & \text{if } \psi > 0 \end{cases}$$

$$(21) \quad \psi = G \text{ on } \delta\Omega$$

One of the characteristics of the solutions of equation (20) with a Dirichlet boundary condition is that: if ψ is solution of the equation, with $\psi = G$ on $\delta\Omega$, then $\varphi = \alpha\psi$ is also a solution with a value $= \alpha G$ on the boundary $\delta\Omega$. In other words, the surface defined by $\psi = 0$ (the plasma cross section) is independant of the boundary value on the shell surface.

After our change of variable we get

$$(22) \quad \left[-\left(\frac{\delta^2 \psi}{\delta x^2} + \frac{1}{x} \frac{\delta^2 \psi}{\delta y^2}\right) = \begin{cases} \lambda \left(1 - \frac{K}{2x}\right) (\psi + G) & \text{if } \psi + G \leq 0 \\ 0 & \text{if } \psi + G > 0 \end{cases} \right.$$

$$(23) \quad \left[\psi = 0 \text{ on } \delta\Omega \right.$$

The weak formulation is then

$$\left[a(\psi, w_j) = \begin{cases} \int_{\Omega} \lambda \left(1 - \frac{K}{2x}\right) (\psi + G) w_j \, dx \, dy & \text{if } \psi + G \leq 0 \\ 0 & \text{if } \psi + G > 0 \end{cases} \right.$$

$$\left[\psi = 0 \text{ on } \delta\Omega \right.$$

Again the problem is non linear as the right hand side is only piecewise linear. However, it has a trivial solution: if the boundary value G is taken positive, then $\psi = 0$ is a trivial solution. We obtained this solution easily.

In order to get one of the other solution, we tested 2 iterations methods:

First iteration method. As before we fix the minimum value of

$$\psi + G$$

$$G^n = G^{n-1} - \min \psi^n + \min \psi^{n-1}$$

$$a(\psi^{n+1}, w_J) = \begin{cases} \lambda \int_{\Omega} \left(1 - \frac{K}{2X}\right) (\psi^n + G^n) w_J \, dx \, dy & \text{if } \psi^n + G^n < 0 \\ 0 & \text{if } \psi^n + G^n > 0 \end{cases}$$

The initial guess is also taken as before, as a single pyramid over the domain. One should be aware of the initial boundary value G^0 and the guessed solution ψ^0 , as the vacuum region could disappear after a small number of iterations.

If the vacuum region disappears, that means we found $G < 0$ at each iteration, and we have merely solved a linear equation.

But increasing the parameter λ seems to keep a vacuum region along with the iterations. Acting on the initial guess alone does not keep automatically a vacuum region along with the iterations.

Figure (2) shows flux surfaces of the nontrivial solution in the rectangular domain shown below and the triangulation of the domain done after 3 subdivisions of the initial triangulation (4 triangles) shown on the picture below

The final triangulation has 256 triangles, 165 nodes (113 interior nodes).

Initial guess was also chosen as a single pyramid over the domain equals to 1.6 at the center of the domain and the initial value of G was chosen equal to 0.6. So the minimum of the real flux was kept fixed to -1 ($= \min \psi^0 + G^0$).

With the parameters $\lambda = 20$, $K = 1$ we get a relative error of the linearization process

$$\frac{\sum_{l=1}^{ns} |y^{n+1}(l) - y^n(l)|}{\sum_{l=1}^N |y^n(l)|}$$

(if $y(l)$ is the value of the solution at each node) of $8 \cdot 10^{-4}$ after 12 iterations. The boundary value is then $G = 0.4629$. The CPU time was 14 sec including the triangulation of the domain and the computation of the total toroidal current $I = 6.467$. The substituting of the finite elements program were in machine language excluding those of the triangulation of the specific domain here a rectangle.

Figure (3) shows the flux surfaces of the nontrivial solution in a circular domain of radius 1 and center coordinates 1.5 at 0, and the triangulation of the domain after 3 subdivisions of the initial triangulation of the domain shown below.

This triangulation has also 256 triangles and 145 nodes.

The initial guess was the same as in the rectangle case with $\psi_{\min}^0 = -1.6$ at $G^0 = 0.6$ and so the minimum of the physical flux was kept equal -1. The parameter λ was 30 and K equals 1.

Figure (4) shows the flux surfaces in the same domain as before but after 4 subdivisions of the initial triangulation that means a computation on 1024 triangles, 565 nodes and 481 interiors nodes. (K equals 25 or $\lambda = 20$ both gives a divergence of this method). Refining the grid gives a smoother solution but we did not notice a real difference in the relative error obtained as both triangulations gave error of $8 \cdot 10^{-3}$ after 15 iterations. The minimum of w was there -1.851 and $G = 0.851$.

Figure (5) are flux surfaces on a D-Shape. The initial triangulation was done with 6 triangles.

and the computation was done after 3 subdivisions on 384 triangles and 217 nodes. $\lambda = 25$ $K = 1$. After 2mn the relative error was still $5 \cdot 10^{-2}$. The minimum of ψ was 1.437 and $G = 0.437$.

2. Second iteration method:

In this section, in order to get the non-trivial solution, instead of fixing the minimum of the flux function and letting the boundary value float at each iteration, we tested a method with fixed boundary value, fixed total toroidal current, and allowed the parameter λ of equation (20) to vary at each iteration of the linearization process.

After normalization, the total toroidal current is given by

$$I = \iint_{\Omega} \frac{L(\Phi)}{r} dr dz$$

$$= \begin{cases} \iint_{\Omega} -\lambda \left(r - \frac{K}{r} \right) \Phi dr dz & \text{if } \Phi < 0 \\ 0 & \text{if } \Phi > 0 \end{cases}$$

We must note here a partial uniqueness result published by Temam [11], as follows:

If we seek a solution $\hat{\Psi}$ to the following problem

$$\begin{aligned} L(\hat{\Psi}) &= -\lambda (r^2 - K) \hat{\Psi} && \text{in the plasma} \\ &= 0 && \text{in the vacuum} \\ \hat{\Psi} &= 0 && \text{on the plasma boundary} \\ G & && \text{on the shell} \end{aligned}$$

$$I = \iint_{\Omega} \frac{L(\hat{\Psi})}{r} dr dz$$

$$c^2 = \iint_{\Omega} -(r^2 - K) \hat{\Psi}^2 dr dz$$

and $d\hat{\Psi}/d\nu$ (the outward normal derivative) continuous on the plasma boundary,

Then, there exists a real $\tau > 0$, such that for all I and c satisfying $I/c < \tau$, the above problem has a unique solution. Thus, we get a unique solution if, for a fixed current I , the constant c is large enough, or if, for a fixed c , I is small enough. (Note that $\lambda = \iint_{\Omega} \left(\frac{K}{r} - r \right) \hat{\psi} dr dz$)

With our flux function ψ which is zero on the boundary, ($\psi = \hat{\psi} - G$), and the variables x and y where $x = r^2/2$ and $y = \sqrt{2} z$,

$$I = \begin{cases} -\frac{\lambda}{\sqrt{2}} \iint_{\Omega} \left(1 - \frac{K}{2x} \right) (\psi + G) dx dy & \text{if } \psi + G \leq 0 \\ 0 & \text{if } \psi + G \geq 0 \end{cases}$$

This constraint occurred in a natural way in the finite element method, as this above integral for the total total toroidal current is very easily calculated; since, at each step, the right hand side of the linear system requires the calculation of the integrals

$$\iint_{\Omega} \left(1 - \frac{k}{2x}\right) (\psi + G) w_J(x, y) dx dy$$

For the current the integration is over the whole domain Ω i.e. over all the triangles of the triangulation, while the above integral is over the support of w_J .

We assign a given triangle to the vacuum region if the values of $\psi + G$ at each vertex of the triangle are ≥ 0 , and so the contribution of this triangle will be zero in the calculation of the current.

If we denote by I the fixed toroidal current, G the fixed value on the boundary, the iteration technique is then:

$$\left\{ \begin{array}{l} I^n = \begin{cases} -\frac{\lambda^n}{J^2} \int_{\Omega} \left(1 - \frac{k}{2x}\right) (\psi^n + G) dx dy & \text{if } \psi^n + G < 0 \\ 0 & \text{if } \psi^n + G > 0 \end{cases} \\ \lambda^{n+1} = \frac{I}{I^n} \lambda^n \\ a(\psi^{n+1}, w_J) = \begin{cases} \lambda^{n+1} \int_{\Omega} \left(1 - \frac{k}{2x}\right) (\psi^n + G) w_J & \text{if } \psi^n + G < 0 \\ 0 & \text{if } \psi^n + G > 0 \end{cases} \end{array} \right.$$

Figure (6) shows the non trivial solution in the rectangular domain

The flux surfaces have been computed with a grid of 113 interior nodes. The fixed current was equal to 6. The value on the boundary $G = 0.6$, $K = 1$. The initial guess is $\psi^0(x,y) = -1.6 w_1(x,y)$ i.e. one single pyramid over the domain. The initial λ^0 has 20. We get relative error

$$\frac{\sum_{l=1}^N |\psi^{n+1}(l) - \psi^n(l)|}{\sum |\psi^n(l)|} < 8 \cdot 10^{-4} \text{ after 59 iterations.}$$

At the limit, the parameter λ was 25.4.

The CPU time including the triangulation of the domain and the drawing of the pictures is 89.13 seconds.

The subroutines, others than those used to the particular triangulation of the rectangle and procedures used for the graphs were in machine language.

The relaxation method at each step used 20 iterations. As a feed back we tried the first iteration method for $\lambda = 25$. Fixing the minimum is faster as we get the same relative error after 14 iterations the minimum was kept fixed to -1. At the limit the $\psi_{\min} = -1.5928$ and so the value of the boundary is 0.5928. We then compute the current and found 6.014. The CPU time was then 14 sec 84 for the same including the triangulation of the domain with the same part of the deck in machine language.

Figure (7) shows the flux surfaces in the case.

BIFURCATION	PHENOMENA	DEPENDING
ON	THE	BOUNDARY
		CONDITION

VALUE $G =$
BOUNDARY CONDITION

△ Finite element method

□ Finite difference method

DOMAIN OF
BIFURCATED
SOLUTIONS

BIFURCATION POINT

RECTANGULAR DOMAIN

$$\beta = 0.5 \quad P(4) = e^{-4}$$

$$\begin{cases} L4 = \beta r^2 e^{-4} \\ 4 = G \quad \text{on } \partial\Omega \end{cases}$$

$$L\psi = \begin{cases} -\lambda(r^2 - K)\psi & \text{if } \psi < 0 \\ 0 & \text{if } \psi > 0 \end{cases}$$

$$\lambda = 20$$

$$K = 1$$

VACUUM CALCULATION

$$\lambda = 30 \quad K = 1$$

NUMBER OF SUBDIVISIONS 3

VACUUM CALCULATION

$$\lambda = 30$$

$$k = 1$$

NUMBER OF SUBDIVISIONS = 4

VACUUM CALCULATION

D shape

Axis

METHOD OF FIXED CURRENT

CURRENT = 5 $\rightarrow \lambda = 25.4$

METHOD OF FIXED MINIMUM

$$\lambda = 25$$

IV. Description of the Program

FINITE ELEMENT PROGRAM

THE FOLLOWING PROGRAM DESCRIBES THE FINITE ELEMENT METHOD OF CODE 1 WITH AUTOMATIC TRIANGULATION APPLIED TO THE NON LINEAR GRAD-SHAFRANOV EQUATION. THE SUBROUTINES DOTRI, DIVTRI, SEGVOT, RESTRI AND TRICER DO THE AUTOMATIC TRIANGULATION OF THE DOMAIN. ONLY TRICER HAS TO BE CHANGED ACCORDING TO THE SHAPE OF THE BOUNDARY. HERE IT IS A CIRCULAR DOMAIN.

VARIABLE AND DIMENSION DESCRIPTION

NS TOTAL NUMBER OF NODES
 NSI TOTAL NUMBER OF INTERIOR NODES
 NT TOTAL NUMBER OF TRIANGLES
 NSEG TOTAL NUMBER OF SEGMENTS
 NSEGI TOTAL NUMBER OF INTERIOR SEGMENTS
 NOTV DESIRED NUMBER OF SUBDIVISIONS OF THE INITIAL TRIANGULATION
 C(2,NS) COORDINATES OF EACH NODE
 NU(3,NT) DESIGNATED NUMBER OF EACH VERTEX IN A TRIANGLE
 NSEGV(NS) NUMBER OF NEIGHBORING SEGMENTS OF EACH POINT AND IS ASSUMED TO BE LESS THAN 6
 NUSEGV(6,NS) DESIGNATED NUMBER OF EACH SEGMENT PASSING THROUGH A NODE
 ISEG(2, NSEG) DESIGNATED NUMBER OF THE END POINTS OF A GIVEN SEGMENT
 JSEG(2, NSEG) DESIGNATED NUMBER OF THE TRIANGLES ADJACENT TO ONE SEGMENT
 KSEG(2, NSEG) DESIGNATED NUMBER OF THE ORIGIN OF A SEGMENT WHEN YOU FOLLOW THE CONTOUR OF THE TRIANGLE COUNTERCLOCKWISE
 IP1(3) ARE THE PERMUTATIONS OF 1,2,3
 NAP(NS-NSI) EQUALS 0 IF THE BOUNDARY CONDITION IS OF DIRICHLET TYPE 1 IF NOT
 IOPT OUTPUT PARAMETER
 =1 PRINT NT, NS, NSI, NSEG OF THE FINAL TRIANGULATION
 =2 PRINTS C(2,NS)
 =3 PRINTS NU(3,NT)
 EPSREL CONVERGENCE CRITERION NUMBER OF THE RELAXATION METHOD
 EPSOM CONVERGENCE CRITERION NUMBER OF THE OPTIMAL RELAXATION PARAMETER
 DIA(NS) DIAGONAL ELEMENTS OF THE MATRIX
 (BILINEAR FORM OF TWO EQUAL BASIS FUNCTIONS WI)
 A(2, NSEG) OTHER ELEMENTS OF THE MATRIX
 (BILINEAR FORM OF TWO DIFFERENT BASIS FUNCTIONS WI AND WJ)
 R(NS) IS THE RIGHT HAND SIDE VECTOR OF THE APPROXIMATE PROBLEM
 Y(NS) IS THE SOLUTION OF THE PROBLEM AT EACH ITERATION OF THE LINEARIZATION PROCESS.

THE PRESSURE IS SUPPOSED QUADRATIC INSIDE THE PLASMA REGION 0 IN THE VACUUM REGION

DIMENSION NAP(48), R(217), X(217), AXX(384), AYY(384), AXY(384),
 IAYX(384), AO(384), A(2,600), DIA(217), C(2,217), NU(3,384), NSEGV(217),
 2NUSEGV(6,217), ISEG(2,600), JSEG(2,600), KSEG(2,600), IP1(3),
 3NU(217,6), Y(169)
 DIMENSION VAL(10)
 REAL LAMBDA
 P=1.
 CY=1.5
 CY=0.
 ITIM=IACCTM(IN)
 INPUT OF THE INITIAL TRIANGULATION OF THE DOMAIN
 CALL DOTRI(NT,NS,NSI,C,NU)
 NOTV=2
 NSEGV=6
 IOPT=1
 LAMBDA=20.
 Q=1.

EPSOM=0.1
 EPSREL=0.001
 AUTOMATIC TRIANGULATION OF THE DOMAIN
 DO 9 K=1, NOTV
 CALL SEGVOT(NT,NS,NSI, NSEG, NSEGI, NU, NSEGV, NUSEGV,
 ISEG, JSEG, KSEG)
 CALL DIVTRI(NT,NS,NSI,C,NU, ISEG, JSEG, KSEG)

```

CALL TRICER(NS,NSI,C,P,CX,CY)
CONTINUE
CALL SEGVPT(INT,NS,NSI,NSEG,NSEGI,NU,NSEGM,NSEGV,
NISEGV,ISEG,JSEG,KSEG)
CALL PESTRI(INT,NS,NSI,NSEG,NSEGI,C,NU,IPT)
GRAPHIC OF THE FINAL TRIANGULATION
CALL TTTT1120)
CALL DWINDO(0.45,2.55,-1.1,1.1)
CALL TWINDO(100,700,100,700)
CALL MAILI(NSEG,C,ISEG)
CALL FRONTNSEGI,NSEG,C,ISEG)
CALL NEWPAS
CALCULATION OF THE ELEMENTS OF THE MATRIX
CALL AIRTRI(INT,C,NU,AXX)
CALL COMATP(INT,C,NU,AYY)
DO 10 J=1,NT
  AXY(J)=0.
  AYY(J)=0.
  AO(J)=0.
10 CONTINUE
CALL MATRI1(INT,NS,NSEG,NSEGI,C,NU,ISEG,JSEG,KSEG,
IAXX,AYY,AXY,AYX,AO,DIA,A)
HOMOGENEOUS DIRICHLET-BOUNDARY-CONDITIONS
  NSJ=NSI+1
  DO 20 I=NSJ,NS
    NAP(I-NSI)=0
20 CONTINUE
THE VALUES OF THE FOLLOWING ARRAY NUT ARE THE DESIGNATED NUMBERS OF THE
TRIANGLES HAVING I AS A VERTEX
  DO 70 J=1,NSI
    N=NSEGV(I)
    DO 30 J=1,N
      NUT(I,J)=0
30 CONTINUE
  II=NSEGV(1,I)
  NUT(I,1)=JSEG(1,II)
  NUT(I,2)=JSEG(2,II)

  K=2
100 K=K+1
  NP1=N+1
  IF (K.EQ.NP1) GO TO 70
  DO 200 J=2,N
    LL=NSEGV(J,I)
    NUT(I,K)=JSEG(1,LL)
    KM1=K-1
    DO 300 M=1,KM1
      IF (NUT(I,M).EQ. NUT(I,K)) GO TO 500
300 CONTINUE
    GO TO 100
  DO 400 M=1,KM1
    IF (NUT(I,M).EQ. NUT(I,K)) GO TO 200
  DO 70 CONTINUE
  GO TO 100
70 CONTINUE
  DO 71 I=NSJ,NS
    N=NSEGV(I)
    NM1=N-1
    DO 72 J=1,NM1
      NUT(I,J)=0
72 CONTINUE
  II=NSEGV(1,I)
  NUT(I,1)=JSEG(1,II)
  K=1
73 K=K+1
  IF (K.EQ.N) GO TO 71
  DO 75 J=2,NM1
    II=NSEGV(J,I)
    NUT(I,K)=JSEG(1,II)
    KM1=K-1

```



```

22 24 M=1,KM1
IF (NUT(I,M).EQ. NUT(I,K)) GO TO 77
76 CONTINUE
GO TO 73
77 IF (JSEF(1,LL).EQ. JSEF(2,LL)) GO TO 79
NUT(I,K)=JSEF(2,LL)
DO 78 M=1,KM1
IF (NUT(I,M).EQ. NUT(I,K)) GO TO 75
78 CONTINUE
GO TO 73
79 NUT(I,K)=JSEF(2,II)
75 CONTINUE
71 CONTINUE
INITIAL GUESS FOR THE MINIMUM VALUE AND BOUNDARY VALUE
XI=-1.6
WRITE(6,1) XI
YMIN=XI
G=0.6
NMN=0
INITIALIZATION OF THE SOLUTION
DO 600 I=1,NSI
X(I)=0.
601 CONTINUE
ITERATION INVOLVING THE LINEARIZATION PROCESS
100 NMN=NMN+1
YMIN1=YMIN

CALCULATION OF THE RIGHT HANDSIDE OF THE LINEAR SYSTEM
DO 1005 I=1,NSI
B(I)=0.
N=NSFV(I)
DO 1003 NA=1,N
IT=NUT(I,NA)
IA=NU(1,IT)
IB=NU(2,IT)
IC=NU(3,IT)
A1=C(1,IA)
A2=C(2,IA)
B1=C(1,IB)
B2=C(2,IB)
C1=C(1,IC)
C2=C(2,IC)
D=(C1-A1)*(A2-B2)-(A1-B1)*(C2-A2)
ALFAIC=(A2-B2)/D
BETAIC=(B1-A1)/D
GAMAIIC=(B2*A1-B1*A2)/D
D=(A1-B1)*(B2-C2)-(B1-C1)*(A2-B2)
ALFAIA=(B2-C2)/D
BETAIA=(C1-B1)/D
GAMAIA=(C2*B1-C1*B2)/D
D=(B1-C1)*(C2-A2)-(C1-A1)*(B2-C2)
ALFAIR=(C2-A2)/D
BETAIR=(A1-C1)/D
GAMAIR=(A2*C1-A1*C2)/D
CALCULATION OF THE BASIS FUNCTIONS WI AND WJ ON EACH TRIANGLE COMPOSING
THE SUPPORT OF WI
LOCATION OF I IN THE TRIANGLE ABC
D1=C(1,I)
D2=C(2,I)
IF (ABS(A1-D1) .LE. 1.E-4 .AND. ABS(A2-D2) .LE. 1.E-4 ) GO TO 11
IF (ABS(B1-D1) .LE. 1.E-4 .AND. ABS(B2-D2) .LE. 1.E-4 ) GO TO 12
ALFAP=ALFAIC
BETAP=BETAIC
GAMAP=GAMAIIC
GO TO 1001
11 ALFAP=ALFAIA
BETAP=BETAIA
GAMAP=GAMAIA
GO TO 1001
12 ALFAP=ALFAIR
BETAP=BETAIR

GAMAP=GAMAIR
600 CONTINUE

```

```

11 (111) = 0.1 * 0.001 * 0.3) GO TO 1001
GO TO 2002
INITIAL DYNAMICAL GUESS
001 IF (IT.GE.1.AND.IT.LE.64) GO TO 2003
IF (IT.GE.65.AND.IT.LE.128) GO TO 2004
IF (IT.GE.129.AND.IT.LE.192) GO TO 2005
IF (IT.GE.193.AND.IT.LE.256) GO TO 2006
003 ALFA1=-YI
BETA1=YI
GAMA1=2.5*XI
X(I)=C(1,I)*ALFA1+C(2,I)*BETA1+GAMA1
GAMA1=GAMA1+G
GO TO 2010
004 ALFA1=-XI
BETA1=-XI
GAMA1=2.5*XI
X(I)=C(1,I)*ALFA1+C(2,I)*BETA1+GAMA1
GAMA1=GAMA1+G
GO TO 2010
005 ALFA1=YI
BETA1=-XI
GAMA1=-0.5*XI
X(I)=C(1,I)*ALFA1+C(2,I)*BETA1+GAMA1
GAMA1=GAMA1+G
GO TO 2010
006 ALFA1=XI
BETA1=XI
GAMA1=-0.5*XI
X(I)=C(1,I)*ALFA1+C(2,I)*BETA1+GAMA1
GAMA1=GAMA1+G
GO TO 2010
VALUE OF THE SOLUTION OVER THE TRIANGLE IA,IB,IC
002 ALFA1=X(IA)*ALFAIA+X(IB)*ALFAIB+X(IC)*ALFAIC
BETA1=Y(IA)*BETAIA+X(IB)*BETAIB+X(IC)*BETAIC
GAMA1=X(IA)*GAMAIA+X(IB)*GAMAIB+X(IC)*GAMAIC
GAMA1=GAMA1+G
010 DO 7000 LL=1, NSI
Y(LL)=X(LL)
000 CONTINUE
V A C U U M T E S T
IF ((X(I)+G).GE.1.E-4) GO TO 1005
TEST FOR TRIANGLES TO HAVE SIDES PARALLEL TO THE AXIS
IF (ABS(A1-C1).LE.1.E-4) GO TO 110
IF (ABS(A1-B1).LE.1.E-4) GO TO 120
IF (ABS(C1-B1).LE.1.E-4) GO TO 130
AC=(A2-C2)/(A1-C1)
ACP=(A1*C2-A2*C1)/(A1-C1)
AB=(A2-B2)/(A1-B1)
ABP=(A1*B2-A2*B1)/(A1-B1)
BC=(B2-C2)/(B1-C1)
BCP=(B1*C2-B2*C1)/(B1-C1)
B(I)=B(I)+FV(Q,A1,C1,AB,ABP,AC,ACP,ALFA1,BETA1,GAMA1,
1,ALFAP,BETAP,GAMAP)+FV(Q,C1,B1,AB,ABP,BC,BCP,ALFA1,BETA1,GAMA1,
2,ALFAP,BETAP,GAMAP)
GO TO 1003
110 AC=(A2-C2)/(A1-C1)
BCP=(B1*C2-B2*C1)/(B1-C1)
AB=(A2-B2)/(A1-B1)
ABP=(A1*B2-A2*B1)/(A1-B1)
B(I)=B(I)+FV(Q,C1,B1,AB,ABP,BC,BCP,ALFA1,
1,BETA1,GAMA1,ALFAP,BETAP,GAMAP)
GO TO 1003
120 AC=(A2-C2)/(A1-C1)
ACP=(A1*C2-A2*C1)/(A1-C1)
BC=(B2-C2)/(B1-C1)
BCP=(B1*C2-B2*C1)/(B1-C1)
B(I)=B(I)+FV(Q,A1,C1,BC,BCP,AC,ACP,ALFA1,BETA1,
1,GAMA1,ALFAP,BETAP,GAMAP)
GO TO 1003

```

```

120 AC=(A2-C2)/(A1-C1)
    ACP=(A1*C2-A2*C1)/(A1-C1)
    AB=(A2-B2)/(A1-B1)
    ABP=(A1*B2-A2*B1)/(A1-B1)
    R(I)=R(I)+FV(Q,A1,C1,AB,ABP,AC,ACP,ALFA1,BETA1,
    GAMMA1,ALFAP,BETAP,GAMAP)
003 CONTINUE
    B(I)=LAMRDA*B(I)
005 CONTINUE
    DO 1004 I=NSJ,NS
        B(I)=0.
004 CONTINUE
    CALCULATION OF THE OPTIMAL RELAXATION PARAMETER
    CALL OMEGA1(NS,NSI,NSEGV,NSEGV,NUSEGV,ISEG,NAP,
    IDIA,A,X,OMEGA,EPSCM,NIT)
    DO 40 I=1,NS
        X(I)=0.
40 CONTINUE
    OVER-RELAXATION METHOD
    CALL RELAX1(NS,NSI,NSEGV,NSEGV,NUSEGV,ISEG,NAP,
    IDIA,A,B,X,OMEGA,EPSPCL,NIT)
    YMIN=0.
    DO 402 I=1,NSI
        YMIN=AMIN1(YMIN,X(I))
402 CONTINUE
    G=G-YMIN+YMIN1
    WRITE(6,1) G
    RELATIVE ERROR
    E=0.
    ERRE=0.
    DO 7001 I=1,NSI
        ERRE=ERRE+ABS(Y(I)-X(I))
        E=E+ABS(X(I))
001 CONTINUE
    ERRE=ERRE/E
    WRITE(6,1) ERRE
    NTIM=(ITIM-IACCTM(IN))*0.000026
    IF(ERRE.LE.1.E-4.OR.NTIM.GE.120) GO TO 501
    GO TO 1100
501 WRITE(6,3) ITIM
    WRITE(6,1) YMIN
    GRAPH OF THE FLUX SURFACES
    NBVAL=9
    DO 602 I=2,NBVAL
        VAL(I)=YMIN-I*YMIN/10.
602 CONTINUE
    CALL DWINDO(0.45,2.55,-1.1,1.1)
    CALL TWINDO(100,700,100,700)
    CALL FRON(NSEGI,NSEG,C,ISEG)
    CALL ISOVAL(NT,C,NU,X,NBVAL,VAL)
    CALL FINITT(0.760)
1   FORMAT(1X,E12.4,/)
3   FORMAT(1X,I4)
    STOP
    END
    SUBROUTINE DOTRI(NT,NS,NSI,C,NU)
    *****
    INITIAL TRIANGULATION
    DIMENSION C(2,145),NU(3,256)

10   READ(5,10)NT,NS,NSI
    FORMAT(3I5)
    WRITE(6,11)NT,NS,NSI
11   FORMAT(1H1,'INITIAL TRIANGULATION',/,1X,22('='),//,
    11X,'NT=',I5,5X,'NS=',I5,5X,'NSI=',I5,/)
    READ(5,20)((C(K,I),K=1,2),I=1,NS)
20   FORMAT(8F8.4)
    WRITE(6,21)((I,(C(K,I),K=1,2),I=1,NS)
21   FORMAT(/,1X,'NODES COORDINATES',/,100(15,6X,
    1F10.4,2X,F10.4,/)
    READ(5,30)((NU(K,J),K=1,2),J=1,NT)
30   FORMAT(18I4)
    WRITE(6,31)(J,(NU(K,J),K=1,2),J=1,NT)
1   FORMAT(/,1X,'VERTICES NUMBERS',/,100(15,6X,I4,2X,
    1I4,2X,I4,/)
    RETURN
    END
    SUBROUTINE DIVTPI(NT,NS,NSI,C,NU,ISEG,JSEG,KSEG)
    DIVISION OF A TRIANGULATION
    DIMENSION C(2,145),NU(3,256),ISEG(2,400),JSEG(2,400),KSEG(2,400)
    DIMENSION IDP(2)

```



```

      IP1(1)=2
      IP1(2)=3
      IP1(3)=1
      N0=(3*(NT+NS)-NSI)/2
      NIO=13*(NT+NSI)-NSI/2
      NF=NS-NSI
      NSEG=(3*NT+NF)/2
      NSEGI=NSEG-NF
      IO=NIO-NSI
      DO 10 I=1,NF
      C(1,NIO+I)=C(1,NSI+I)

      C(2,NIO+I)=C(2,NSI+I)
10  CONTINUE
      DO 100 L=1,NSEGI
      I1=ISEG(1,L)
      I2=ISEG(2,L)
      J1=JSEG(1,L)
      J2=JSEG(2,L)
      K1=KSEG(1,L)
      K2=KSEG(2,L)
      KK1=IP1(K1)
      KK2=IP1(K2)
      J11=4*(J1-1)+K1
      J12=4*(J1-1)+KK1
      J13=4*J1
      J21=4*(J2-1)+K2
      J22=4*(J2-1)+KK2
      J23=4*J2
      NU(K1,J11)=I1
      NU(KK2,J22)=I1
      IF(I2.GT.NSI) I2=I2+IO
      NU(K2,J21)=I2
      NU(KK1,J12)=I2
      I3=NSI+L
      NU(KK1,J11)=I3
      NU(K1,J12)=I3
      NU(K1,J13)=I3
      NU(KK2,J21)=I3
      NU(K2,J22)=I3
      NU(K2,J23)=I3
      C(1,I3)=0.5*(C(1,I1)+C(1,I2))
      C(2,I3)=0.5*(C(2,I1)+C(2,I2))
100 CONTINUE
      NSEGI=NSEGI+1
      DO 200 L=NSEGI,NSEG
      I1=ISEG(1,L)
      I2=ISEG(2,L)
      J=JSEG(1,L)
      K=KSEG(1,L)
      KK=IP1(K)
      J1=4*(J-1)+K
      J2=4*(J-1)+KK
      J3=4*J
      I1=I1+IO
      I2=I2+IO

      NU(K,J1)=I1
      NU(KK,J2)=I2
      I3=NIO+NF+L-NSEGI
      NU(KK,J1)=I3
      NU(K,J2)=I3
      NU(K,J3)=I3
      C(1,I3)=0.5*(C(1,I1)+C(1,I2))
      C(2,I3)=0.5*(C(2,I1)+C(2,I2))
200 CONTINUE
      NT=4*NT
      NS=IO
      NSI=NIO
      RETURN
      END

```

SUBROUTINE SEGVAL (NT, NS, NSI, ISEG, NSEG, NU, NSEGV,
NSEGV, NUSEGV, ISEG, JSEG, KSEG)

NEIGHBORING SEGMENTS OF A NODE

DIMENSION NU(3,256), NSEGV(145), NUSEGV(6,145), ISEG(2,400),
JSEG(2,400), KSEG(2,400)
DIMENSION IPI(3)

IPI(1)=2

IPI(2)=3

IPI(3)=1

DO 10 I=1, NS

NSEGV(I)=0

DO 10 K=1, NSEGM

NUSEGV(K, I)=0

10 CONTINUE

NSEG=NT+NS-1

NSEGI=NSEG-(NS-NSI)

NUSI=0

NUSF=NSEGI

DO 20 L=1, NSEG

DO 20 K=1, 2

ISEG(K, L)=0

JSEG(K, L)=0

KSEG(K, L)=0

20 CONTINUE

DO 30 J=1, NT

DO 30 K=1, 3

K1=IPI(K)

I1=NU(K, J)

I2=NU(K1, J)

IF (I1.GT.NSI.AND.I2.GT.NSI) GO TO 31

27 IF (I1.GT.I2) GO TO 30

NUSI=NUSI+1

NUS=NUSI

GO TO 32

31 DO 29 JJ=1, NT

IF (JJ.EQ.J) GO TO 29

DO 28 KK=1, 3

I1=NU(KK, JJ)

IF (I1.NE.I1) GO TO 28

I11=NU(IPI(KK), JJ)

I12=NU(IPI(IPI(KK)), JJ)

IF (I11.EQ.I1.OR.I12.EQ.I1) GO TO 27

28 CONTINUE

29 CONTINUE

33 NUSF=NUSF+1

NUS=NUSF

32 NSEGV(I1)=NSEGV(I1)+1

NUSEGV(NSEGV(I1), I1)=NUS

ISEG(1, NUS)=I1

JSEG(2, NUS)=I2

NSEGV(I2)=NSEGV(I2)+1

NUSEGV(NSEGV(I2), I2)=NUS

30 CONTINUE

DO 40 J=1, NT

DO 40 K=1, 3

K1=IPI(K)

I=NU(K, J)

I1=NU(K1, J)

NSEGVJ=NSEGV(I1)

DO 40 KK=1, NSEGVJ

L=NUSEGV(KK, I)

IF (ISEG(2, L).EQ.I1) GO TO 50

IF (JSEG(1, L).EQ.I1) GO TO 60

GO TO 40

50 JSEG(1, L)=J

KSEG(1, L)=K

GO TO 40

60 JSEG(2, L)=J

KSEG(2, L)=K

0 CONTINUE

NSEGVJ=NSEGVJ+1

DO 70 L=NSEGVJ, NSEG

JSEG(2, L)=JSEG(1, L)

KSEG(2, L)=KSEG(1, L)

70 CONTINUE

RETURN

END

SUBROUTINE PESTRI (NT, NS, NSI, NSEG, NSEGI, C, NU, IOPT)

FINAL TRIANGULATION

DIMENSION C(2,145), NU(3,256)

WRITE(6,10)

10 FORMAT(1H1, 'FINAL TRIANGULATION ', /, 1X, 20('*'), /)

WRITE(6,20) NT, NS, NSI, NSEG, NSEGI

20 FORMAT(1X, 'NUMBER OF TRIANGLES', /, 1X, 'NUMBER OF NODES', /, 1X, 'NUMBER OF INTERIOR NODES', /, 1X, 'NUMBER OF SEGMENTS', /, 1X, 'NUMBER OF INTERIOR SEGMENTS', /, 1X,

3/)

IF(IOPT.EQ.1) GO TO 70

WRITE(6,30)

30 FORMAT(1H1, 'NODES COORDINATES ', /, 1X, 22('*'), /)

WRITE(6,40) (1, (C(K,I), K=1,2), I=1, NS)

40 FORMAT(4(15, 2F5.3, 4X))

IF(IOPT.EQ.2) GO TO 70

WRITE(6,50)

50 FORMAT(1H1, 'VERTICES NUMBERS OF THE TRIANGLES', /, 1X,

33('*'), /)

WRITE(6,60) (J, (NU(K,J), K=1,3), J=1, NT)

60 FORMAT(4(15, 2X, 3I5, 2X))

70 RETURN

END

SUBROUTINE FRON(NSEGI, NSEG, C, ISEG)

GRAPH OF THE BOUNDARY

DIMENSION C(2,145), ISEG(2,400)

NSEGI=NSEGI+1

DO 10 I=NSEGI, NSEG

X1=C(1, ISEG(1, I))

X2=C(2, ISEG(1, I))

CALL MOVEA(X1, X2)

X1=C(1, ISEG(2, I))

X2=C(2, ISEG(2, I))

CALL DRAWA(X1, X2)

10 CONTINUE

RETURN

END

SUBROUTINE TRICER(NS, NSI, C, P, CX, CY)

CIRCLE TRIANGULATION

TRANSFORMATION OF THE BOUNDARY POINTS

DIMENSION C(2,145)

NSI=NSI+1

DO 10 I=NSI, NS

R1=(CX-C(1, I))**2+(CY-C(2, I))**2

R1=SQRT(R1)

IF(R1.GT.0.999*R) GO TO 10

R2=R/R1

C(1, I)=C(1, I)*R2+CX*(1.-R2)

C(2, I)=C(2, I)*R2+CY*(1.-R2)

10 CONTINUE

RETURN

END

SUBROUTINE MATLL(NSEG, C, ISEG)

GRAPH OF THE FINAL TRIANGULATION

DIMENSION C(2,145), ISEG(2,400)

DO 10 I=1, NSEG

X1=C(1, ISEG(1, I))

X2=C(2, ISEG(1, I))

CALL MOVEA(X1, X2)

X1=C(1, ISEG(2, I))

X2=C(2, ISEG(2, I))

CALL DRAWA(X1, X2)

10 CONTINUE

RETURN

END

SUBROUTINE APTPI(NT, C, NU, AXX)

AREA OF EACH TRIANGLE AFTER TRIANGULATION

DIMENSION C(2,145), NU(3,256), AXX(256)

```

DO 10 J=1,NT
  I1=NU(1,J)
  I2=NU(2,J)
  I3=NU(3,J)
  C11=C(1,I1)
  C12=C(1,I2)
  C13=C(1,I3)
  C21=C(2,I1)
  C22=C(2,I2)
  C23=C(2,I3)
  AXX(J)=0.5*(C11*C22-C12*C21+C12*C23-C13*C22+C13*C21-
  C11*C23)
10 CONTINUE
  RETURN
END
SUBROUTINE CCMATR(NT,C,NU,AXX)
  C
  INTEGRAL OF 1/X OVER THE TRIANGLE NUMBER J
  DIMENSION C(2,145),NU(3,256),AXX(256)
  DO 10 J=1,NT
    IA=NU(1,J)
    IB=NU(2,J)
    IC=NU(3,J)
    A1=C(1,IA)
    A2=C(2,IA)
    B1=C(1,IB)
    B2=C(2,IB)
    C1=C(1,IC)
    C2=C(2,IC)
    IF(A1 .EQ. C1) GO TO 110
    IF (A1 .EQ. B1) GO TO 120
    IF (C1 .EQ. B1) GO TO 130
    AC=(A2-C2)/(A1-C1)
    ACP=(A1*C2-A2*C1)/(A1-C1)
    AB=(A2-B2)/(A1-B1)
    ABP=(A1*B2-A2*B1)/(A1-B1)
    RC=(B2-C2)/(B1-C1)
    BCP=(B1*C2-B2*C1)/(B1-C1)
    AYY(J)=(AC-AB)*(C1-A1)+(ACP-ABP)*ALOG(C1/A1)
    AYY(J)=AXX(J)+(RC-AB)*(B1-C1)+(BCP-ABP)*ALOG(B1/C1)
    GO TO 10
  110 RC=(B2-C2)/(B1-C1)
    BCP=(B1*C2-B2*C1)/(B1-C1)
    AB=(A2-B2)/(A1-B1)
    ABP=(A1*B2-A2*B1)/(A1-B1)
    AYY(J)=(RC-AB)*(B1-C1)+(BCP-ABP)*ALOG(B1/C1)
    GO TO 10
  120 AC=(A2-C2)/(A1-C1)
    ACP=(A1*C2-A2*C1)/(A1-C1)
    RC=(B2-C2)/(B1-C1)
    BCP=(B1*C2-B2*C1)/(B1-C1)
    AYY(J)=(AC-BC)*(C1-A1)+(ACP-BCP)*ALOG(C1/A1)
    GO TO 10
  130 AC=(A2-C2)/(A1-C1)
    ACP=(A1*C2-A2*C1)/(A1-C1)
    AB=(A2-B2)/(A1-B1)
    ABP=(A1*B2-A2*B1)/(A1-B1)
    AYY(J)=(AC-AB)*(C1-A1)+(ACP-ABP)*ALOG(C1/A1)
10 CONTINUE
  RETURN
END
SUBROUTINE MATRI1(NT,NS,NSEG,NSEGI,C,NU,ISEG,JSEG,KSEG,
  IAXX,AXX,AYX,AYX,A0,DIA,A)
  C
  NON ZERO MATRIX ELEMENTS OF AN ELLIPTIC PROBLEM OF ORDER 2,
  C
  DISCRETIZED BY FINITE ELEMENT METHOD OF ORDER 1 ON AN AUTOMATIC
  C
  TRIANGULATION
  DIMENSION C(2,145),NU(3,256),ISEG(2,400),JSEG(2,400),KSEG(2,400),
  IAXX(256),AXX(256),AYX(256),A0(256),DIA(145),A(2,400)
  DIMENSION IP1(3)
  IP1(1)=2
  IP1(2)=3
  IP1(3)=1
  DO 10 I=1,NS
    DIA(I)=0
  10 CONTINUE

```

```

      K=1, 1
      I=NU(K, J)
      K1=IP1(K)
      K2=IP1(K1)
      I1=NU(K1, J)
      I2=NU(K2, J)
      AA=C(2, I1)-C(2, I2)
      BB=C(1, I2)-C(1, I1)
      DIA(I)=DIA(I)+(C.25/(AXX(J)**2))*(AA*AA*AXX(J)+BB*BB*
1  AYY(J)+AA*BB*(AXY(J)+AYX(J)))+AO(J)/6.
20  CONTINUE
      DO 30 L=1, NSEG
      A(1, L)=0.
      A(2, L)=0.
30  CONTINUE
      DO 40 L=1, NSEG
      DO 50 K=1, 2
      IF (L.GT.NSEG1.AND.K.EQ.2) GO TO 50
      I=ISEG(K, L)
      J=JSEG(K, L)
      KK=KSEG(K, L)
      I1=NU(IP1(KK), J)
      I2=NU(IP1(IP1(KK)), J)
      AA1=C(2, I1)-C(2, I2)
      AA2=C(2, I2)-C(2, I)
      BB1=C(1, I2)-C(1, I1)
      BB2=C(1, I)-C(1, I2)
      CXY=AA2*BB1

      CXX=AA1*BB2
      COF=0.25/(AXX(J)**2)
      TT=COF*(AA1*AA2*AXX(J)+BB1*BB2*AYY(J))+AO(J)/12.
      TXY=COF*(CXY*AXY(J)+CYX*AYX(J))
      TYX=COF*(CYX*AXY(J)+CXY*AYX(J))
      IF (K.EQ.2) GO TO 60
      A(1, L)=TT+TXY
      A(2, L)=TT+TYX
      GO TO 50
60  A(1, L)=A(1, L)+TT+TXY
      A(2, L)=A(2, L)+TT+TYX
50  CONTINUE
40  CONTINUE
      RETURN
      END
      FUNCTION FV(K, A1, C1, AB, ABP, AC, ACP, ALFA, BETA, GAMA, ALFAP, BETAP,
1  GAMAP)
      AUXILIARY INTEGRAL FOR THE COMPUTATION OF THE RIGHT HANDSIDE
      INTEGRAL OF (1-K/2X)*(ALFA*X+BETA*Y+GAMA)*(ALFAP*X+BETAP*Y+GAMAP)
      OVER Y BETWEEN AB*X+ABP AND BC*X+BCP THEN INTEGRATED OVER X FROM A1 TO B1
      REAL K
      AA=BETA*BETAP/3.
      BB=AC+AB
      CC=ACP+ABP
      AAA=(ALFA*BETAP+ALFAP*BETA)/2.
      BBB=(GAMA*BETAP+BETA*GAMAP)/2.
      CCC=ALFAP*GAMA+ALFA*GAMAP
      P=AA*(AC*AC+AC*AB+AB*AB)
      P=P+AAA*PB+ALFA*ALFAP
      Q=AA*(2.*AC*ACP+AB*ACP+AC*ABP+2.*AB*ABP)
      Q=Q+BBB*PB+AAA*CC+CCC
      R=AA*(ACP*ACP+ACP*ABP+ABP*ABP)
      R=R+BBB*CC+GAMA*GAMAP
      ACB=AC-AB
      ACPB=ACP-ABP
      A=P*ACB
      B=P*ACBP+Q*ACB
      C=Q*ACBP+R*ACP
      D=ACBP*R
      A12=A1*A1
      A13=A12*A1
      A14=A13*A1
      C12=C1*C1
      C13=C12*C1
      C14=C13*C1
      F=A*(C14-A14)/4.
      F=F+(2.*B-K*A)*(C13-A13)/6.
      F=F+(2.*C-K*B)*(C12-A12)/4.
      F=F+(2.*D-K*C)*(C1-A1)/2.
      A1=C17A1
      F=F-K*D*ALOG(AC1)/2.
      FV=F
      RETURN

```

```

SUBROUTINE OMEGA(NS,NSI,NSEGM,NSEGV,NUSEGV,ISEG,NAP,
  IDIA,A,X,OMEGA,EPSOM,NIT)
OPTIMAL RELAXATION PARAMETER USED IN THE OVER RELAXATION METHOD
  DIMENSION NSEGV(145),NUSEGV(6,145),ISEG(2,400),NAP(32),DIA(145),
  IA(2,400),X(145)

```

```

DO 10 I=1,NS
  IF(I.GT.NSI) GO TO 11
12 X(I)=1.
  GO TO 10
11 IF(NAP(I-NSI).EQ.1) GO TO 12
  X(I)=0.
10 CONTINUE
  NIT=NIT+1
  PCMA=1000.
  POMI=1000.
  DO 20 I=1,NS
    IF(I.LE.NSI) GO TO 21
    IF(NAP(I-NSI).EQ.0) GO TO 20
21 XI=X(I)
    S=0.
    NSEGVJ=NSEGV(I)
    DO 30 K=1,NSEGVJ
      L=NUSEGV(K,I)
      I1=ISEG(1,L)
      I2=ISEG(2,L)
      IF(I1.EQ.I2) GO TO 31
      S=S+A(I1,L)*X(I2)
      GO TO 30
31 S=S+A(I2,L)*X(I1)
30 CONTINUE
    X(I)=-S/DIA(I)
    QUO=X(I)/XI
    PCMA=AMAX1(PCMA,QUO)
    POMI=AMIN1(POMI,QUO)
20 CONTINUE
    PCMA=AMIN1(PCMA,0.999999)
    QMI=2./(1.+SQRT(1.-PCMI))
    QMA=2./(1.+SQRT(1.-POMA))
    IF((PCMA-QMI).GT.EPSOM.AND.NIT.LT.100) GO TO 15
    OMEGA=(QMI+QMA)/2.
    RETURN
END

```

```

SUBROUTINE RELAX1(NS,NSI,NSEGM,NSEGV,NUSEGV,ISEG,NAP,
  IDIA,A,B,X,OMEGA,EPSREL,NIT)
  DIMENSION NSEGV(145),NUSEGV(6,145),ISEG(2,400),NAP(32),DIA(145),
  IA(2,400),B(145),X(145)
  SOLUTION BY OVER RELAXATION METHOD OF THE LINEAR SYSTEM IN A FINITE
  ELEMENT METHOD OF ORDER 1

```

```

  NIT=0
10 NIT=NIT+1
  S1=0.
  S2=0.
  DO 20 I=1,NS
    IF(I.LE.NSI) GO TO 21
    IF(NAP(I-NSI).EQ.0) GO TO 20
21 XI=X(I)
    S=0.
    NSEGVJ=NSEGV(I)
    DO 30 K=1,NSEGVJ
      L=NUSEGV(K,I)
      I1=ISEG(1,L)
      I2=ISEG(2,L)
      IF(I1.EQ.I2) GO TO 31
      S=S+A(I1,L)*X(I2)
      GO TO 30
31 S=S+A(I2,L)*X(I1)
30 CONTINUE
    X(I)=(-S+B(I))/DIA(I)
    X(I)=XI+OMEGA*(X(I)-XI)
    S1=S1+ABS(XI-X(I))
    S2=S2+ABS(X(I))
20 CONTINUE
    DRM=S1/S2
    IF(DRM.GT.EPSREL) GO TO 10
  WRITE(6,50) NIT
50 FORMAT(10X,15,/)
  RETURN
END

```

```

SUBROUTINE ISOVAL(NT,C,NU,X,NBVAL,VAL)
GRAPH OF THE FLUX SURFACES
DIMENSION C(2,145),NU(3,256),X(145),VAL(10)
DIMENSION FF(3),COORD(2,3),IP1(3)

```

```

IP1(1)=2
IP1(2)=3
IP1(3)=1
DO 20 J=1,NT
DO 11 K=1,3
I=NU(K,J)
FF(K)=X(I)
COORD(1,K)=C(1,I)
COORD(2,K)=C(2,I)

```

```

11 CONTINUE
FFF1=FF(1)
FFF2=FF(2)
FFF3=FF(3)
D12=ABS(FFF1-FFF2)
D23=ABS(FFF2-FFF3)
FFMA=AMAX1(ABS(FFF1),ABS(FFF2),ABS(FFF3))
IF(D12+D23.LT.AMAX1(0.00001,0.0001*FFMA)) GO TO 20
DO 26 NVAL=1,NBVAL
VAL1=VAL(NVAL)
ITP=0
DO 27 K=1,3
FEK=FF(K)
FEK1=FF(IP1(K))
FMI=AMIN1(FEK,FEK1)
FMA=AMAX1(FEK,FEK1)
DIF=FMA-FMI
IF(DIF.LT.0.00001) GO TO 27
EPS=0.001*DIF
IF(VAL1.LT.(FMI-EPS).OR.VAL1.GT.(FMA+EPS)) GO TO 27
HH=ABS(FEK-VAL1)/DIF
FF1=COORD(1,K)+HH*(COORD(1,IP1(K))-COORD(1,K))
FF2=COORD(2,K)+HH*(COORD(2,IP1(K))-COORD(2,K))
IF (ITP.EQ. 0) GO TO 23
CALL DRAWA(FF1,FF2)
GO TO 27
28 ITP=1
CALL MOVEA(FF1,FF2)
27 CONTINUE
26 CONTINUE
20 CONTINUE
RETURN
END

```

APPENDIX TO THE PROGRAM

Formulas which gives the values of NS, NT, NSI, NSEG, NSF after NDiV subdivisions from the values NS_0 , NT_0 , NSI_0 of the initial triangulation.

$$NT = (4)^{NDiV} NT_0$$

$$NSF = 2^{NDiV} (NS_0 - NSI_0)$$

number of boundary
points

$$NSEG = (3 \times NT + NSF)/2$$

$$NS = NSEG - NT + 1$$

$$NSI = NS - NSF$$

ACKNOWLEDGEMENTS

I wish to thank Professor C.K. Chu who made my stay at Columbia University possible and fruitful.

I wish to thank Dr. M. Reusch for his helpful suggestions.

I wish to thank J.F.Bourgat from IRIA (Paris) for making available the program of finite element he developed.

REFERENCES

- ¹P.G. Ciarlet, M.H. Schultz, R.S. Vaiga, "Numerical methods of high order accuracy for non linear boundary value problems." Numer. Math 13-51-77 (1969).
- ²K. Yosida, Functional Analysis - Springer Verlag Berlin (1965).
- ³Courant and Hilbert, Methods of Mathematical Physics - Vol I Interscience, New York (1953).
- ⁴Shafranov V.D., Review of Plasma Physics (Leontovich MA ED.) 2. Consultant, Bureau - New York(1966).
- ⁵Marder and Weitzner, A bifurcation Problem in E- layer Equilibria, Plasma Physics 12, 434-445.
- ⁶M.S. Berger, A bifurcation theory for non linear Elliptic Partial differential Equations and Related Systems (p 113-216). Bifurcation theory and Non linear Eigenvalue Problems J.B. Keller and Autman. Editors. W.A. Benjamin, Inc. Publisher, New York (1969).
- ⁷J.B. Keller, Some positive Problems Suggested by non linear heat generation p 217-256. Bifurcation theory and non linear Eigenvalue problems J.B. Keller and Autman. Editors, W.A. Benjamin, Inc. Publishers, New York (1969).
- ⁸M.F. Reusch, Numerical Solution of the Axisymmetric Magneto-hydrodynamic equilibrium equation. PhD thesis- Columbia University May 1976.
- ⁹Y. Suzuki, Free Boundary MHD-Equilibria in Axisymmetric tori, Nuclear Fusion 14, 365-382 (1976).

¹⁰Feinberg-Lackner, Multipole tokamak Equilibria, Nuclear Fusion 13, 569 (1973).

¹¹Temam, R., A nonlinear equivalence problem: the shape at equilibrium of a confined plasma Arch Rat Mech Anal , 60 (1975) n° 1.