180
3-7-80

180. 801

Report No. 80
COO-2456-69
UC-20 g

Columbia University
in the City of New York

A NUMERICAL STUDY OF THE INITIAL DYNAMICS IN TORMAC

by

Ahmet Y. Aydemir

MASTER

1979

Research supported by DOE Contract EY-76-S-02-2456.

Plasma Laboratory

School of Engineering and Applied Science

Columbia University

New York, N.Y. 10027

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

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.

A NUMERICAL STUDY OF THE INITIAL DYNAMICS IN TORMAC

Ahmet Y. Aydemir

DISCLAIMER

This book 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.

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Faculty of Pure Science

COLUMBIA UNIVERSITY

1979

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ABSTRACT

A NUMERICAL STUDY OF THE INITIAL DYNAMICS IN TORMAC

Ahmet Y. Aydemir

Using a single fluid magnetohydrodynamic (MHD) code, we study the initial plasma dynamics in Tormac, an experimental device which attempts to confine plasma in a toroidal, two-pole cusp field.

The code is in two dimensions and includes classical electrical conductivity and a constant heat conductivity. The full set of nonlinear MHD equations for the mass density ρ , fluid velocity $\underline{u}$, temperature T , the poloidal flux function ψ , and the poloidal current function χ are solved using a "predictor-corrector" method; an explicit step for $\underline{u}$ is followed by an alternating-direction-implicit (ADI) step, first for ψ , and then χ , T , and finally for ρ , and $\underline{u}$ again.

The absence of a conducting surface enclosing the device makes it difficult to solve for the flux function ψ . We first investigate the use of Green's function techniques in which at each time step, one makes use of a "virtual conducting shell", and then tries to subtract the effects of surface currents induced on the shell using the known Green's function for this geometry. This method, widely used in ideal MHD equilibrium calculations, proves to be expensive in our initial-boundary value problem. In the present version of the code, we use a mapping technique where the upper half of the infinite poloidal plane is mapped

onto a finite domain in a (ρ, ζ) coordinate system, where $\rho = 1/(r+a)$ and $\zeta = 1/(z+b)$ (a and b are constants). The vacuum equation for ψ is solved in this coordinate system.

The results of our simulations indicate that the external fields in Tormac are such that, starting with some reasonable initial conditions, the plasma never reaches the desired equilibrium state. It initially forms two horn-like structures attached to a main body of plasma, which help the plasma get further away from the desired configuration. Eventually, most of the plasma ends up near the inner wall, with a temperature in the range of 10 electron volts.

To improve the plasma behavior, we try various changes in the initial conditions and boundary values. As long as we stay within reasonable bounds, changes in the initial conditions do not affect the final plasma state to a great extent. Altering the boundary values, however do lead to improvements. We present a succession of changes in the external current distribution and timing which markedly improve the initial dynamics. With the elimination of some of the currents, and delaying in time of some of the others, we are able to implode the plasma to a configuration very similar to the desired cusp equilibrium. However, after this initial phase, the plasma is not in static equilibrium. It flows along the open poloidal field lines towards the cusp region, and the top and bottom walls of the device. This flow is supposed to be impeded by the mirroring effect of converging field lines in these regions, but we show that the plasma is too cold, and thus too collisional, to support any pressure anisotropy necessary for such

mirror confinement.

Finally, using a heuristic argument we explain the reconnection of the poloidal field lines in our simulations. Initially closed field lines open up at a rate much faster than expected from resistive diffusion. We show that this process, similar in nature to coalescing of two magnetic islands, can be explained in terms of an argument Kadomtsev uses in his explanation of non-linear development of the $m=1$ resistive internal kink mode.

ACKNOWLEDGEMENTS

I wish to express my thanks to my thesis advisor Prof. C. K. Chu for his support, guidance and understanding during the course of this research.

I am also grateful to Prof. R. A. Gross for his moral support and help.

Many special thanks go to Dr. H. C. Lui for his help during my "breaking-in" period at Columbia, and whose belt-pinch code was the starting point for my Tormac code.

I also thank Dr. Paul Weber for many fruitful discussions, Dr. Ben Feinberg, who introduced the problem to us, and Drs. Mike Vella, Morton Levine, Herbert Berk, and the rest of the Tormac group at LBL for their help during this research.

Finally I thank my wife Eunsung for being so understanding, and my parents for their unconditional moral support. This work is dedicated to them.

This work was supported by the United States Department of Energy under contract EY-76-S-02-2456.

TABLE OF CONTENTS

	Page
Abstract	i
Acknowledgements	iv
I. INTRODUCTION	1
II. FORMULATION OF THE PROBLEM	12
II.1 The Plasma Region	12
II.2 The Vacuum Region	19
II.3 Boundary Conditions on Plasma-Vacuum Boundary	20
II.4 Initial Conditions	29
II.5 Solution Procedure for the Flux Function	33
III. NUMERICAL METHODS	39
III.1 Formulation of the Finite Difference Equations	39
III.2 Numerical Boundary Conditions	47
III.3 Low Density Region and Artificial Viscosity	49
III.4 Solution for the Poloidal Flux Function	51
III.4a Mapping of the Infinite Domain to a Finite Domain	63
III.4b The Solution Algorithm and the Finite Difference Equations	71
III.4c Numerical Solution of the Vacuum Equation	78
IV. RESULTS	80
IV.1 Analysis of the Results	91
IV.2 Numerical Experiments	103

	Page
IV.2a Effects of Various Initial Conditions	103
IV.2b Higher External Fields	106
IV.2c Changes in the External Current Distribution and Timing	107
IV.2d Motion Along the Open Field Lines	119
IV.3 Disappearance of the Closed Poloidal Field Lines	123
IV.3a A Heuristic Argument for the Rate of Reconnection	126
REFERENCES	138
APPENDICES	
I. Implementation of the Continuity Conditions on the Plasma-Vacuum Boundary	142
II. Cholesky Decomposition and Conjugate-Gradient Iterations	145
III. Computer Program Listing	157

TABLE OF FIGURES

I.1, I.2	2
I.3, I.4	9
II.1	13
II.2, II.3	24
II.4	31
II.5	34
II.6	35
II.7	36
III.1	40
III.2	53
III.3	54
III.4	61
III.5	64
III.6	65
III.7	67
IV.1	81
IV.2	82
IV.3	83
IV.4	84
IV.5	86
IV.6	87
IV.7	88
IV.8	89
IV.9	90
IV.10	92
IV.11	94
IV.12, IV.13	96
IV.14	98
IV.15	101
IV.16	102
IV.17	105
IV.18	109
IV.19	110
IV.20	111
IV.21	112
IV.22, IV.23	115
IV.24	116
IV.25	117
IV.26	118
IV.27	120
IV.28	121
IV.29	125
IV.30, IV.31	128
IV.32	133
IV.33	134
IV.34	135
A2.1	150

I. INTRODUCTION

The confinement schemes in magnetic fusion may be roughly divided into two groups. In the first group the plasma is tied to the field lines which fill a closed toroidal volume. In the second group the plasma is still tied to the field lines, but here the field lines leave the plasma volume, and the confinement is through the mirroring effect on the "open" field lines. Tokamaks are the best known example of the devices from the first group, where the field lines ergodically cover nested toroidal flux surfaces. Mirror machines, of course, fall into the second category with open field lines.

Tormac is in a way a hybrid between these two categories, having both open and closed field lines in a toroidal geometry. Rather than starting with the complicated Tormac field geometry and trying to explain the function of the various field components, we will attempt to trace out the history of Tormac, first starting with a simple cusp on which the basic Tormac idea is based, and slowly build up towards it by gradually adding the various field components.

The simplest cusp geometry, a spindle cusp, is produced by two circular coils carrying currents in opposite directions (Fig. I.1). Such geometries were considered for plasma confinement in the early days of the magnetic fusion program [1], mainly because of their absolute magnetohydrodynamic (MHD) stability, even against perturbations of finite magnitude [2]. This is a consequence of the fact that everywhere on the plasma boundary the curvature of the field lines points away from the plasma volume, except at the cusps, which are singular points across which the curvature changes sign discontinuously; the plasma sits in a

FIG. I.1 A SPINDLE CUSP

FIG. I.2 A TOROIDAL FOUR-POLE CUSP

potential well with the potential increasing along every outward normal to the boundary. Therefore, any perturbation of the plasma surface leads to a higher energy state, thus stabilizing the perturbations.

The main problem with these cusp geometries has always been the rapid loss of plasma through the cusps. An important parameter determining the loss rate is the effective "hole" size at the cusps through which the particles are lost. Various estimates of the radius of the hole put it anywhere between an electron gyroradius to many ion gyroradii, a value of a few ion gyroradii being the widely accepted one. Assumptions and models used in arriving at these estimates can be found in the review articles by Spalding [3] and Haines [4]. The fact that the spindle cusp is very "leaky" can be explained through the free flow of particles out through the ring cusp and the two point cusps present in a spindle cusp geometry. We can estimate the particle confinement time in this geometry as follows:

Assuming a hole size of one ion gyroradius r_i at the ring cusp, the total area A through which particles escape is given by

$$A = 2\pi r_i R_p$$

where R_p is the radius of the ring (See Fig. 1.1). Then the average total particle flux is

$$\Gamma = Anv_i = 2\pi r_i R_p n v_i$$

where n is the number density and v_i is the average ion thermal speed. The plasma volume goes like R^3 . Then the particle confinement time is

$$\tau \sim \frac{nR_p^3}{\Gamma} = \frac{R_p^2}{2\pi r_i v_i} \sim \frac{R_p^2 B}{T_i} \quad (I.1)$$

Note that τ scales like the Bohm diffusion time ! Confinement time determined by this relation for thermonuclear plasmas is quite short, which is why the spindle cusp geometries have been largely ignored in magnetic fusion research.

Not all cusp geometries have such severe loss rates. Levine, et. al. [5] show that in a toroidal axisymmetric line cusp (Fig. I.2), conserved generalized momentum p_ϕ , where ϕ is the ignorable coordinate in this geometry, reduces the loss rate by preventing "some ions from aligning their velocity vectors closely enough with the magnetic field direction to go through the larger mirror-like field which in practical systems exists above the cusp lines". In other words the generalized momentum p_ϕ ,

$$p_\phi = mR^2 \dot{\phi} + \frac{q}{c} R A_\phi$$

where A_ϕ is the toroidal component of the vector potential, defines a loss cone in velocity space such that some ions which have the proper orientation are lost immediately through the line cusps, whereas the others have to "wait" for a collision to scatter them into this loss cone.

In a subsequent paper Boozer and Levine [6] show that one can improve upon the geometry of Fig. I.2 by "stuffing" the cusp with a toroidal field, which is actually an old idea due to Berkovitz, et. al. [1]. Addition of this toroidal field improves the confinement time in two ways [7]. First of all, it divides the particles into two groups:

a) The particles on the plasma-vacuum boundary (sheath) with their loss-cone distribution in velocity space because of the canonical invariant p_ϕ , and

b) the interior particles which "see" only the toroidal field and are kept away by it from the sheath.

Thus we have the following picture: The particles in the sheath are mirror confined. They scatter into the loss cone in an ion-ion collision time τ_i and are lost from the system. The losses from the sheath are compensated by the influx of particles from the big reservoir of "internal particles", which make it to the sheath only through cross field diffusion in the presence of the toroidal field. Note that here we are ignoring the effects of curvature, gradient, and $E \times B$ drifts, etc. possible in a toroidal geometry.

We can estimate the particle confinement time in this stuffed cusp geometry as follows [7]: Particle confinement time in the sheath is τ_i , an ion-ion collision time, i.e. the sheath is emptied in a time equal to τ_i . Then the total confinement time is equal to $m\tau_i$, where m is the number of times the sheath can be replenished from the interior reservoir. Assuming that the plasma radius is R_p , and the sheath thickness is r_i , an ion gyroradius, $m \sim R_p/r_i$. Thus, the confinement time is given by

$$\tau \sim \frac{R_p}{r_i} \tau_i \quad (I.2)$$

Writing τ_i as λ_i/v_i , where λ_i is the ion mean free path, and v_i is the ion thermal velocity, we have

$$\tau \sim \frac{R_p}{r_i} \frac{\lambda_i}{v_i} = \frac{R_p^2}{r_i v_i} \frac{\lambda_i}{R_p} = \tau_{sp} \frac{\lambda_i}{R_p}$$

where τ_{sp} is given by Eq. (I.1), and is equal to the confinement time for a spindle-cusp. Note that the confinement time for a stuffed-cusp is better by a factor of λ_i/R_p .

The second advantage of stuffing the cusp has to do with the existence of an adiabatic (as opposed to exact canonical) invariant, which was originally introduced by Grad [8] and Grossmann [9]. With the addition of the toroidal field, the magnetic field is not zero anywhere in the plasma, and the adiabatic invariant

$$J = \oint p_{\perp} dx_{\perp}$$

where $p_{\perp}$ and $x_{\perp}$ are momentum and position vectors perpendicular to the magnetic field, respectively, is defined everywhere. This further restricts a particle's motion and reduces the size of the loss cone. In summary, the canonical invariant p_{ϕ} , which exists in an axisymmetric toroidal line cusp even without a toroidal field, introduces a loss cone, which is made even smaller by the adiabatic invariant J ; more particles are trapped in the sheath and can escape only after collisions which break the conservation of p_{ϕ} , and J , the canonical and adiabatic invariants, respectively.

So far we have started with a spindle cusp with $\tau \sim R_p^2/r_i v_i$ and arrived at a toroidal stuffed line cusp geometry with a confinement time better by a factor of λ_i/R_p . At this point we have to answer two questions. First of all, at what parameter regime is $\lambda_i/R_p \gg 1$, i.e. for what temperature and density values is the mean free path

long enough to give us an improvement in the confinement time ? Secondly, what are the deleterious effects of a pure toroidal field in the interior region ?

Using classical expressions for collision frequency ν_i and ion thermal velocity v_i [10], for λ_i we obtain

$$\lambda_i = 2 \times 10^{12} T_i^2 / n \quad (\text{cm.})$$

Assuming a plasma radius, $R_p = 5 \text{ cm.}$, and $n = 10^{15}$ particles per cc, we have

$$\frac{\lambda_i}{R_p} = 4 \times 10^{-4} T_i^2$$

Thus to have $\lambda_i / R_p > 1$ we need at least $T_i > 50 \text{ eV}$, i.e. we need to heat the plasma at least up to 50 eV before we can benefit from this "improved" confinement scheme.

The answer to second question is well known. A pure toroidal field in the interior region will introduce gradient and curvature drifts, which in turn will lead to an $E \times B$ drift in the plasma, resulting in the deterioration of the confinement characteristics.

Hammer and Kunkel [11] consider a macroscopic rotation of the plasma in the poloidal plane about the minor axis as a possible cure for the problems introduced by the toroidal field and conclude that "slow mass rotation about the minor axis is probably permissible in Tormac without causing difficulties". Of course, a second cure is to introduce a magnetic rotational transform by inducing a toroidal current in the interior region, as in a Tokamak, which may actually be easier than inducing a bulk plasma rotation in an experimental device. In fact,

this is the method adopted in Tormac IV experiment, where a toroidal current produces closed field lines in the plasma region.

The addition of this poloidal field completes our historical journey from the spindle cusp to Tormac (TORoidal Magnetic Cusp) at Lawrence Berkeley Laboratory. Tormac is a stuffed toroidal line cusp [12]. Because of certain technical advantages and reduced losses [13], instead of the four-pole cusp of Fig. I.2, a bi-cusp configuration is used. In this geometry confinement is by both toroidal and poloidal fields. In the poloidal plane the plasma cross section has a shield shape with two cusp points (Fig. I.3). The toroidal field dominates at small R , and the poloidal (or the cusp) fields dominate at large R . The poloidal field lines follow the contour of the shield-shaped plasma. The relative magnitude of the fields are chosen so as to make the plasma boundary a constant $|B|$ surface. The plasma sits in an absolute min- $|B|$ well and is permeated by a toroidal field (stuffed cusp). It also carries a toroidal current which produces the desired closed poloidal field lines in the plasma. For more information about the design and construction of Tormac IV, the reader is referred to the report by Gallagher and Levine [14]. Fig. I.4 shows the various field profiles at the mid-plane in Tormac (assuming a sharp-boundary plasma). Inside we basically have Tokamak field profiles, which are separated from the confining toroidal and poloidal fields by sheath currents.

At early stages of the experiment there was very little diagnostic information available about the plasma behavior. Some spectroscopic data based on the Stark broadening of H_{β} line, and the Doppler broadening of $HeII$ (4686-Å) line [12,15] were interpreted to imply temperatures of

FIG. 1.3 A TOROIDAL DI-POLE CUSP.

FIG. 1.4. Field Profiles at the mid-plane in a sharp boundary model of Tormac.

well over 100 eV, and a well confined plasma. Spectroscopic data is prone to misinterpretation, and without other supporting density and temperature measurements, one could not be really sure about what exactly was happening in the experiment. In particular, one could not determine whether the plasma was reaching the shield-shaped equilibrium configuration and being confined in that state for any length of time.

Our goal in this research was to numerically simulate the initial dynamics of the plasma. Here "the initial dynamics" is loosely defined as the plasma motion during the period between the end of the preionization stage, during which the fill gas is ionized, and the toroidal bias field and the toroidal current are established, and the time at which the external confining fields reach their maximum values. In particular, we wanted to :

- 1) Start with some reasonable initial conditions that approximate the conditions in the plasma that exist after preionization,
- 2) Apply the external fields as prescribed in the experiment,
- 3) Follow the dynamics of the plasma in time to see if and how it approached equilibrium, and
- 4) See how close that equilibrium state is to the desired shield-shaped equilibrium configuration.

In the next chapter we explain the magnetohydrodynamic (MHD) equations we use to model the plasma. We formulate the exact problem to be solved and discuss the physical initial and boundary conditions. Chapter III describes the numerical methods employed in solving the equations, and the numerical initial and boundary conditions. In Chapter IV we present the main results of our simulations

and show that the plasma does not implode towards the desired equilibrium state. We discuss some "numerical experiments" conducted to solve the problems of the existing Tormac design. We suggest some modifications to the external current distribution and timing which solve some of the problems and make it possible for the plasma to implode to a state which resembles the desired equilibrium configuration. We also discuss the validity of one of our assumptions, namely the assumption of isotropic pressure, in the density and temperature regimes we observe in the simulations, and also present some results about the fast disappearance of the closed poloidal field lines during the implosion, which may negatively affect the confinement and the stability of the internal plasma region in the experiments.

Before we go on to the next chapter, some general remarks about the fate of the Tormac experiments is in order. The experiments have recently been terminated, since Tormac has never come close to achieving its design goals. Thomson scattering measurements [16] have deflated the temperature figures from hundreds of electron volts that were originally claimed down to 5 eV range, and density scans have shown that the plasma, far from reaching the desired equilibrium position, invariably ends up near the inner wall. It seems to us that the Tormac project would have benefited a great deal from a study like ours, if it had been done a few years earlier.

II. FORMULATION OF THE PROBLEM

Starting with some reasonable initial conditions which we will describe later (Sect. II.4), we will prescribe the external currents as functions of time and observe the the initial plasma motion , implosion, bouncing, etc., under the influence of the fields produced by these currents. For our purposes, the toroidal vacuum vessel will determine the boundaries of a toroidal volume of "plasma region"; the infinite space outside this vessel will be the "vacuum region".

The region inside the vacuum vessel is assumed to be filled with plasma at all times; due to numerical difficulties which will be explained in the next chapter, we do not allow a vacuum to develop during the implosion in this region; therefore, one set of equations describing the plasma will be valid in the whole domain within the vessel, and mathematically, the vacuum vessel will be referred to as the "plasma boundary", or the "plasma- vacuum boundary". Of course, the vacuum region outside this boundary will be described by a different set of equations.

Throughout the calculations we will use the cylindrical (r, ϕ, z) coordinate system, but the axisymmetry of the problem will make ϕ an ignorable coordinate (Fig. II.1). Intrinsic 2-dimensionality of the problem does not allow us to further reduce it to 1-D. Thus, we will work in the (r, z) poloidal plane .

II.1 The plasma Region

The plasma will be modelled in terms of the single fluid

FIG. II.1. Vacuum vessel, and the coordinate system used in the calculations.

magnetohydrodynamic (MHD) equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \underline{u}) = 0 \quad (\text{II.1})$$

$$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot (\rho u \underline{u}) = -\nabla p + \frac{1}{c} \underline{J} \times \underline{B} \quad (\text{II.2})$$

$$\rho c_v \left\{ \frac{\partial T}{\partial t} + (\underline{u} \cdot \nabla) T \right\} = -p \nabla \cdot \underline{u} + \nabla \cdot \{ K \nabla T \} + \frac{J^2}{\sigma} \quad (\text{II.3})$$

$$\frac{\partial \underline{B}}{\partial t} = -c \nabla \times \underline{E} \quad (\text{II.4})$$

$$\underline{J} = \frac{c}{4\pi} \nabla \times \underline{B} \quad (\text{II.5})$$

$$\underline{E} + \frac{1}{c} \underline{u} \times \underline{B} = \underline{J} / \sigma \quad (\text{II.6})$$

$$\nabla \cdot \underline{B} = 0 \quad (\text{II.7})$$

where the symbols are defined as follows:

ρ : mass density,

$\underline{u}$: macroscopic fluid velocity,

T : temperature,

p : pressure, $p = (\gamma - 1) \rho c_v T$,

K : coefficient of heat conductivity,

σ : electrical conductivity,

c_v : specific heat at constant volume,

γ : ratio of specific heats, c_p / c_v .

Eq. II.1 is the usual mass continuity equation. (II.2) is an

equation, written in the conservation form, for the momentum density, with the scalar pressure term ∇p , and the Lorentz term $\underline{J} \times \underline{B}$. The energy equation (II.3) has a source term $\underline{J} \cdot \underline{J} / \sigma$ corresponding to ohmic heating. (II.4) is the usual Faraday's law. The Ampere's law (II.5) is written without the displacement current term. (II.6) is an equation for Ohm's law, and (II.7) states that the magnetic field lines have no sources or sinks. If (II.7) is used as an initial condition, the divergence free nature of the field is ensured for all times by the Faraday's Induction law (II.4).

Note that since we assume $T_i = T_e$, and $n_i = n_e$, we have $p = 2nkT = 2\rho RT = 2(\gamma - 1)\rho C_V T$. The factor of two is absorbed into the constants k , R , and C_V .

Equations (II.4-II.7) can be combined to derive the following vector equation for the $\underline{B}$ fields:

$$\frac{\partial \underline{B}}{\partial t} = \nabla \times (\underline{u} \times \underline{B}) - \frac{c^2}{4\pi} \nabla \times \{ (\nabla \times \underline{B}) / \sigma \} \quad (\text{II.8})$$

For a constant conductivity case, it can be written as

$$\frac{\partial \underline{B}}{\partial t} = \nabla \times (\underline{u} \times \underline{B}) + \frac{c^2}{4\pi\sigma} \nabla^2 \underline{B} \quad (\text{II.8}')$$

which clearly shows the parabolic, diffusive nature of the equation, with a diffusion coefficient equal to $c^2/4\pi\sigma$.

Instead of calculating the B fields directly from (II.8), it is customary and convenient to follow an indirect route. With the axisymmetry condition $\frac{\partial}{\partial \phi} = 0$, $\underline{B}_p$, the poloidal component of $\underline{B}$, can be derived from the toroidal component of the vector potential $\underline{A}$ alone. Then we can write the total $\underline{B}$ as [1]

$$\underline{B} = \nabla \times (\vec{\phi} A_\phi) + \vec{\phi} B_\phi \quad (\text{II.9})$$

Noting that the unit vector $\vec{\phi}$ can be put in the form

$$\vec{\phi} = r \nabla \phi$$

where ϕ is the toroidal angle, we have

$$\underline{B} = \nabla \times \{ \nabla \phi \cdot r A_\phi \} + r B_\phi \nabla \phi$$

Letting $\psi = r A_\phi$, and $\chi = r B_\phi$, we get

$$\underline{B} = \nabla \psi \times \nabla \phi + \chi \nabla \phi \quad (\text{II.10})$$

or in terms of $\vec{\phi}$,

$$\underline{B} = \frac{\nabla \psi \times \vec{\phi}}{r} + \frac{\chi}{r} \vec{\phi} \quad (\text{II.10}')$$

Expanding the $\nabla \psi$ term we have

$$B_r = -\frac{1}{r} \frac{\partial \psi}{\partial z}, \quad B_z = \frac{1}{r} \frac{\partial \psi}{\partial r} \quad (\text{II.11a})$$

Also recall that

$$B_\phi = \frac{\chi}{r} \quad (\text{II.11b})$$

To get the physical meaning of ψ , let us calculate the poloidal flux through a circle of radius r , with the center on the z -axis :

$$F_{\text{pol}} = \int_0^{2\pi} \int_0^r (\underline{B} \cdot \vec{z}) r dr d\phi = 2\pi \int_0^r (\vec{r} \cdot \nabla \psi) dr$$

or

$$F_{\text{pol}} = 2\pi \{ \psi(r,z) - \psi(0,z) \} \quad (\text{II.12})$$

Since B_r is finite ,

$$\lim_{r \rightarrow 0} rB_r = - \lim_{r \rightarrow 0} \frac{\partial \psi}{\partial z} = 0$$

i.e., at $r=0$, ψ is independent of z . Thus, $\psi(0,z) = \text{constant}$ in Eq. II.12, which can be set equal to zero without affecting the poloidal field B_{p} , since B_{p} is determined by derivatives of ψ . Then

$$F_{\text{pol}} = 2\pi\psi$$

i.e., $\psi(r,z)$ is proportional to the actual poloidal flux at the point (r,z) .

By noting the parallel between $\psi = rA_{\phi}$, and $\chi = rB_{\phi}$ we see that χ is probably proportional to the total poloidal current through a circle of radius r at the point (r,z) , which we can prove as follows. Taking the curl of Eq. (II.10'), we have

$$J_{\phi} = -\frac{c}{4\pi} \left\{ \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \psi}{\partial r} \right) + \frac{1}{r} \frac{\partial^2 \psi}{\partial z^2} \right\} \quad (\text{II.13})$$

$$J_r = -\frac{c}{4\pi} \frac{1}{r} \frac{\partial \chi}{\partial z}, \quad J_z = \frac{c}{4\pi} \frac{1}{r} \frac{\partial \chi}{\partial r} \quad (\text{II.14})$$

Then

$$I_z(r,z) = \int_0^{2\pi} \int_0^r J_z r dr d\phi = 2\pi \int_0^r \frac{c}{4\pi} \frac{\partial \chi}{\partial r} dr$$

or

$$I_z(r,z) = \frac{c}{2} [\chi(r,z) - \chi(0,z)] \quad (\text{II.15})$$

Since $\chi = rB_\phi$, $\chi(0,z)$ is identically zero; therefore,

$$\chi = \frac{2}{c} I_z, \quad (\text{II.15}')$$

which proves our point.

The toroidal component of (II.8) gives the following equation for χ [2]:

$$\begin{aligned} \frac{\partial \chi}{\partial t} = & \frac{c^2}{4\pi} \left[r \frac{\partial}{\partial r} \left(\frac{1}{\sigma r} \frac{\partial \chi}{\partial r} \right) + \frac{\partial}{\partial z} \left(\frac{1}{\sigma} \frac{\partial \chi}{\partial z} \right) \right] \\ & - \frac{\partial}{\partial r} \left[\frac{u_r \chi}{r} - u_\phi B_r \right] + \frac{\partial}{\partial z} \left[r u_\phi B_z - u_z \chi \right] \end{aligned} \quad (\text{II.16})$$

The flux function ψ is determined by the equation

$$\frac{\partial \psi}{\partial t} + (\underline{u} \cdot \nabla) \psi = \frac{c^2}{4\pi\sigma} \Delta^* \psi \quad (\text{II.17})$$

where

$$\Delta^* \psi = r \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \psi}{\partial r} \right) + \frac{\partial^2 \psi}{\partial z^2} \quad (\text{II.18})$$

Equations II.1-3, 11,13,14,16,17, with their corresponding initial-boundary conditions, completely determine the variables ρ , $\underline{u}$, T , $\underline{B}$, and $\underline{J}$, which describe the state of the plasma in our magnetohydrodynamic model. The advantages of solving for the fields through (II.16-17), instead of directly from (II.8), are two-fold. First of all, (II.8) is actually 3 equations for the three components

of $\underline{B}$; thus, using this indirect method we reduce the number of equations to be solved down to 2, namely (II.16-17). Secondly, as we will see later, it is easier to specify boundary conditions for ψ , and χ than for B_r , B_ϕ and B_z .

II.2 The Vacuum Region

At the beginning of this chapter we defined the vacuum region as the infinite domain outside the rectangular vacuum vessel of Fig. IV.1. What happens to the fields in this region would ordinarily be quite peripheral to our problem, since we are interested in the plasma dynamics, and not in the details of the field profiles in vacuum outside the plasma region. But the lack of a conducting surface enclosing the plasma region in Tormac makes it impossible to assign boundary values for the flux function ψ on the plasma-vacuum boundary without having any information about the poloidal fields outside the plasma. Thus, we are forced to calculate ψ at least in a limited domain in the vacuum region to be able to find the necessary boundary values in the solution of Eq. II.17 in the plasma region.

The $\underline{B}$ fields in vacuum at any instant in time are determined by the Ampere's law

$$\nabla \times \underline{B} = \frac{4\pi}{c} \underline{J}$$

where $\underline{J}$ is the current density in the external conductors. Substituting for the poloidal field $\underline{B}_p = \nabla\psi \times \nabla\phi$, we obtain

$$\Delta^* \psi = - \frac{4\pi}{c} r J_\phi \quad (II.19)$$

where again $\Delta^* = r \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial}{\partial r} \right) + \frac{\partial^2}{\partial z^2}$, and J_ϕ is the current density in the external toroidal coils carrying the currents I_1 , I_2 , and I_3 (Fig. IV.1). As we will see in the next section, Eq. II.19 is the only one we need in the vacuum region, since the boundary values for χ to be used in the solution of (II.16) can be determined quite easily from the Ampere's law without solving any differential equations.

II.3 Boundary Conditions on Plasma-Vacuum Boundary

The fluid equations II.1-2 for the variables ρ , ρu form a hyperbolic set by themselves. The energy equation (II.3), and the equations for the variables ψ , and χ (Eqs. II.16-17) are parabolic, because of their dissipative (heat-equation like) nature. The hyperbolic set and the parabolic equations are coupled through Eqs. II.11,13,14 and together form a parabolic set.

In the limit of zero resistivity and heat conductivity, the equations reduce to a much-studied hyperbolic set of ideal MHD equations. This set has seven characteristics [3], corresponding to slow, intermediate (Alfven), and fast MHD waves, and one corresponding to fluid velocity u . In the case of plasma confined within a non-porous, conducting boundary, the field lines are parallel to the wall, and only one of these characteristics, namely one of the pair which corresponds to a magnetoacoustic wave, points inward towards the plasma region; thus, only one boundary condition, $u_\perp = 0$, is needed. But non-conducting walls lead to complications in that now the field lines can enter and leave the plasma volume. In that case we would have at least three

characteristics pointing inward and have to specify more boundary conditions than just $u_1 = 0$.

In our non-ideal case where we have finite electrical conductivity and non-zero heat conductivity, we have a completely different problem. Here the general view is that one could treat the hyperbolic set (Eqs. II-1,2) and the parabolic set (Eqs. II.3,16,17) formally separately, even though the whole set becomes parabolic when one of the equations is parabolic. In the case of plasma surrounded by conducting walls, one still could set $u_1 = 0$, and the values of the independent variables, or their normal derivatives would be specified for the parabolic equations. This view, which is based on the incorrect assumption that one could simply carry over from the ideal MHD case merely by adding some more boundary conditions for the parabolic equations as needed, overlooks the fact that in a strict mathematical sense, we do not have the seven characteristics of ideal MHD anymore. Thus, it is not a priori obvious how many boundary conditions are necessary, or whether setting $u_1 = 0$ is a sufficient condition for the fluid equations in the non-ideal case.

To clear this ambiguity in the number of required boundary conditions, we have to "count characteristics", i.e., we have to find the characteristics of the set of parabolic equations we have and determine how many of them are directed inward and thus carry information into the plasma region. We will carry out this task for the following set of equations in 1-D and carry over the results to our 2-D problem:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} = 0 \quad (\text{II.20})$$

$$\rho \left\{ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} \right\} = - \frac{\partial p}{\partial x} - \frac{1}{8\pi} \frac{\partial B^2}{\partial x} \quad (\text{II.21})$$

$$\frac{\partial B}{\partial t} + \frac{\partial (uB)}{\partial x} = \frac{c^2}{4\pi\sigma} \frac{\partial^2 B}{\partial x^2} \quad (\text{II.22})$$

$$\rho c_v \left\{ \frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} \right\} = - p \frac{\partial u}{\partial x} + \frac{J^2}{\sigma} + K \frac{\partial^2 T}{\partial x^2} \quad (\text{II.23})$$

$$J = \frac{c}{4\pi} \frac{\partial B}{\partial x}$$

$$p = (\gamma-1) \rho c_v T$$

where

σ : electrical conductivity (constant),

K : heat conductivity (constant).

The characteristics of this set of equations are curves in the (x,t) plane across which we may have discontinuities in the normal derivatives of the dependent variables [3],[4]. To be able to find such curves, we write the equations in terms of a new coordinate system $\{\xi(x,t), \eta(x,t)\}$:

$$(\xi_t + u \xi_x) \frac{\partial \rho}{\partial \xi} + \rho \xi_x \frac{\partial u}{\partial \xi} + (\text{other terms}) = 0 \quad (\text{II.20}')$$

$$\begin{aligned} \rho (\xi_t + u \xi_x) \frac{\partial u}{\partial \xi} + (\gamma-1) c_v \left\{ \rho \xi_x \frac{\partial T}{\partial \xi} + \xi_x T \frac{\partial \rho}{\partial \xi} \right\} \\ + \xi_x \frac{B}{4\pi} \frac{\partial B}{\partial \xi} + (\text{other terms}) = 0 \end{aligned} \quad (\text{II.21}')$$

$$\xi_x B \frac{\partial u}{\partial \xi} + (\xi_t + u \xi_x - \eta \xi_{xx}) \frac{\partial B}{\partial \xi} \quad (\text{II.22}')$$

$$-\eta \xi_x^2 \frac{\partial^2 B}{\partial \xi^2} + (\text{other terms}) = 0$$

$$\xi_x (\gamma-1) T \frac{\partial u}{\partial \xi} - \frac{\eta}{4\pi\rho c_V} \xi_x^2 \left(\frac{\partial B}{\partial \xi} \right)^2 - \frac{K}{\rho c_V} \xi_x^2 \frac{\partial^2 T}{\partial \xi^2} \quad (\text{II.23}')$$

$$(\xi_t + u \xi_x - \frac{K}{\rho c_V} \xi_{xx}) \frac{\partial T}{\partial \xi} + (\text{other terms}) = 0.$$

"Other terms" contain derivatives of ρ , u , etc., with respect to the variable η . Assuming that $\xi(x,t) = \text{constant}$ is a characteristic curve, the governing equations for the jumps in the derivatives are found by writing the equations for two different curves (see Fig. II.2),

$$\xi(x,t) = \text{constant} + \varepsilon, \text{ and } \xi(x,t) = \text{constant} - \varepsilon,$$

keeping η constant, subtracting the corresponding equations and letting $\varepsilon \rightarrow 0$. For instance, from (II.20') we have

$$\lim_{\varepsilon \rightarrow 0} \{ (\xi_t + u \xi_x)_{+\varepsilon} \rho_\xi|_{+\varepsilon} - (\xi_t + u \xi_x)_{-\varepsilon} \rho_\xi|_{-\varepsilon}$$

$$+ \rho \xi_x u_\xi|_{+\varepsilon} - \rho \xi_x u_\xi|_{-\varepsilon} \} = 0$$

which gives

$$(\xi_t + u \xi_x) [\rho_\xi] + \rho \xi_x [u_\xi] = 0 \quad (\text{II.24})$$

FIG. II.2. Adjacent characteristics used in the derivation of the jump conditions.

FIG. II.3. Four characteristics of the parabolic set in 1-D (Eqs. II.20-23).

where $[\rho_\xi]$ is the jump in $\frac{\partial \rho}{\partial \xi}$ given by

$$\lim_{\epsilon \rightarrow 0} \left\{ \frac{\partial \rho}{\partial \xi} \Big|_{+\epsilon} - \frac{\partial \rho}{\partial \xi} \Big|_{-\epsilon} \right\}$$

Similarly from the other equations we get

$$\rho (\xi_t + u \xi_x) [u_\xi] + (\gamma-1) c_v T \xi_x [\rho_\xi] = 0 \quad (\text{II.25})$$

$$\xi_x B [u_\xi] - \eta \xi_x^2 [B_{\xi\xi}] = 0 \quad (\text{II.26})$$

$$(\gamma-1) \xi_x T [u_\xi] - \frac{K}{\rho c_v} \xi_x^2 [T_{\xi\xi}] = 0. \quad (\text{II.27})$$

In the equations for B and T above, the derivatives $\partial B / \partial \xi$, and $\partial T / \partial \xi$ are assumed to be continuous across the characteristics, since only the discontinuities in the highest order derivatives ($\partial^2 B / \partial \xi^2$ and $\partial^2 T / \partial \xi^2$ above) are important in the determination of the characteristics [4].

The requirement for a non-trivial solution for $[\rho_\xi]$, $[u_\xi]$, $[B_{\xi\xi}]$, and $[T_{\xi\xi}]$ to exist is that $\det \tilde{A} = 0$, where $\tilde{A}$ is the coefficient matrix :

$$\tilde{A} = \begin{pmatrix} (\xi_t + u \xi_x) & \rho \xi_x & 0 & 0 \\ (\gamma-1) c_v T \xi_x & \rho (\xi_t + u \xi_x) & 0 & 0 \\ 0 & \xi_x B & -\eta \xi_x^2 & 0 \\ 0 & (\gamma-1) \xi_x T & 0 & -\frac{K}{\rho c_v} \xi_x^2 \end{pmatrix}$$

Letting $\det \tilde{A} = 0$ gives

$$\xi_x^4 \{ (\xi_t + u \xi_x)^2 - a_0^2 \xi_x^2 \} = 0 \quad (\text{II.28})$$

where $a_0 = \left\{ \frac{\partial p}{\partial \rho} \right\}^{1/2} = \{ (\gamma-1) c_v T \}^{1/2}$, the sound speed. The solution

$$\xi_x^4 = 0$$

corresponds to the four identical characteristics which belong to the two parabolic equations for T and B . (Note that the heat equation $T_t = k T_{xx}$ alone has two identical characteristics given by $\xi_x^2 = 0$). Thus, we need to specify T or $\partial T / \partial n$, and B or $\partial B / \partial n$ on the boundaries. (Of course, for each equation we can specify a mixed boundary condition of the form $\alpha B + \beta \frac{\partial B}{\partial n}$.)

For $\xi_x \neq 0$, we have from Eq. II.28

$$c = -u \pm a_0$$

where $c = \xi_t / \xi_x$. Since $\frac{dx}{dt} = -\xi_t / \xi_x$ on $\xi(x, t) = \text{constant}$ curves, we get the following equation for two additional characteristics:

$$\frac{dx}{dt} = u \pm a_0, \quad a_0 = \left\{ \frac{\partial p}{\partial \rho} \right\}^{1/2} \quad (\text{II.29})$$

Recall that a_0 is simply the sound speed. In going from ideal to non-ideal MHD, we have lost the characteristics corresponding to Alfvén, fast, and slow waves. At a non-porous boundary, only one of the characteristics determined by Eq. II.29 points into the plasma region; therefore, we need to specify only one boundary condition, $u_\perp = 0$, for the continuity and momentum equations, regardless of whether the boundary is conducting or not. Fig. II.3 shows the "four"

characteristics in our 1-D problem.

Now we can carry over these ideas to our more complicated 2-D problem determined by equations (II.1-3, 16, and 17). In analogy with the 1-D problem we see that we will have two characteristics

$$\frac{dx}{dt} = u \pm a_0$$

essentially determined by the mass continuity and the momentum equations, plus one characteristic

$$\xi_x = 0 \quad \left(\text{or } \frac{dt}{dx} = 0 \right)$$

for each of the three parabolic equations for temperature T (Eq. II.3), the flux function ψ (Eq. II.17), and $\chi = rB_\phi$ (Eq. II.16). Thus, we have a total of five characteristics, but at a solid boundary, only one of the two given by $\frac{dx}{dt} = u \pm a_0$ will point into the plasma region; therefore, in a strict mathematical sense we need to specify only four quantities:

$$\begin{aligned} 1) & \quad u_\perp \\ 2) & \quad \alpha_{11} T + \alpha_{12} \frac{\partial T}{\partial n} \\ 3) & \quad \alpha_{21} \psi + \alpha_{22} \frac{\partial \psi}{\partial n} \\ 4) & \quad \alpha_{31} \chi + \alpha_{32} \frac{\partial \chi}{\partial n} \end{aligned} \tag{II.30}$$

where $\hat{n}$ is the normal to the boundary, and either one of the coefficients

$$\alpha_{i1}, \quad \alpha_{i2}, \quad (i=1,2,3)$$

may be zero.

As mentioned earlier $u_\perp = 0$ is a mathematical statement of the fact that we have a solid, non-porous wall around the plasma region,

through which there is no mass flux. Similarly, we may set the heat flux equal to zero ($\partial T / \partial n = 0$, adiabatic wall), or we can specify a constant temperature wall ($T = T_w$).

As it was pointed out earlier in section II.1, $\chi(r, z, t)$ is a function of the total current flowing through a circle of radius r (Fig. II.4); it is constant everywhere on the boundary, even if the boundary is non-conducting, and it is determined by Eq. II.15',

$$\chi_w = \frac{2I_z}{c}$$

Finding the boundary values of ψ turns out to be a little more complicated, especially if the walls are not conductors. But let us first consider the easier case of plasma bounded by a conducting shell. In this case $B_n = 0$ on the walls, which implies the tangential derivative of ψ is zero; thus, ψ_w is constant, i.e., the wall is a flux surface. This constant value of the flux function $\psi(t)$ is specified as a function of time so as to give the correct toroidal current I_ϕ in the plasma [2], [5]. Park [6] uses a more elaborate method in which he starts with a specific current profile $I_\phi(t)$ and tries to adjust $\psi(t)$ continuously to produce that profile in time.

If the boundary is an insulator, as it is in Tormac, ψ is an a priori unknown function of both space and time, and it has to come out of the solution for ψ in the infinite poloidal plane

$$\{ (r, z), 0 \leq r < \infty, -\infty < z < \infty \}.$$

Although we can not assign specific values for ψ on the plasma-vacuum boundary, we may set forth some conditions it has to

satisfy. First of all, ψ has to be continuous across this boundary; otherwise, we would have infinite B fields there. Moreover, the normal derivative of ψ , $\partial\psi/\partial n$ (or the tangential B field) has to be continuous. This is required by the fact that the walls are insulators, and that because of the finite electrical conductivity, the plasma can not support surface currents. Thus, although we do not know beforehand either ψ , or $\partial\psi/\partial n$, we know that both have to be continuous across the boundary:

$$[\psi] = 0 \quad \left[\frac{\partial\psi}{\partial n} \right] = 0 \quad (\text{II.31})$$

II.4 Initial Conditions

Counting characteristics, or from physical reasoning, we see that we have to assign initial conditions for all our dependent variables, ρ , $\underline{u}$, T , ψ , and χ . These initial conditions are determined by the experimental conditions existing in the plasma at the beginning of the period we are simulating.

During the pre-ionization stage in Tormac, the plasma forms a pinch with a small toroidal current, which produces closed poloidal field lines. At the same time the plasma is permeated by a small toroidal bias field. The details of the state of the initial pinch are not known experimentally. Although the initial density and temperature profiles do not seem to affect to a great extent the dynamics later on, as we will show in Chapter IV, we try to approximate this initial pinch in our

simulations by starting at $t=0$ with non-uniform density and temperature profiles (Fig. II.4). The density and temperature in the pinch are typically a few times higher than the respective background values. Since this pinch comes under the influence of the external forces in a very short time during the implosion, we do not attempt to start with an equilibrium configuration. For typical initial conditions, the imbalance in the kinetic and magnetic field pressures is small enough so that during the short time the pinch is free of external influences, the profiles do not change appreciably.

Specifying an initial toroidal current in the pinch poses a small problem in that in the code, the toroidal current density J_ϕ is not one of the primary variables - variables advanced in time by difference equations, but it is calculated from the poloidal flux function ψ ; therefore, we have to specify an initial condition for ψ corresponding to a toroidal current in the pinch. Since we do not expect the time evolution of the plasma to strongly depend on the initial configuration, instead of making an elaborate and exact calculation for the initial value of ψ , we find an approximate value as follows:

In the toroidal coordinate system (r, ϕ, θ) of Fig. II.4, the flux function ψ is related to the toroidal current density J_ϕ by [5]

$$\frac{\partial}{\partial r} \left(\frac{r}{R} \frac{\partial \psi}{\partial r} \right) + \frac{\partial}{\partial \theta} \left(\frac{1}{rR} \frac{\partial \psi}{\partial \theta} \right) = - \frac{4\pi}{c} r J_\phi \quad (\text{II.32})$$

where $R = R_0 + r \cos \theta$, and R_0 is the major radius. In terms of the non-dimensional quantities

$$\tilde{r} = r/r_0, \quad \tilde{R} = R/r_0, \quad \tilde{\psi} = \psi / (r_0^2 B_0), \quad \tilde{J}_\phi = J_\phi / \left(\frac{c B_0}{4\pi r_0} \right)$$

FIG. II.4. Location of the initial pinch ($r < r_0$), and the (r, ϕ, θ) coordinate system used in the calculation of the initial ψ .

where r_0 is the minor radius of the region in which J_ϕ flows, and B_0 is some characteristic field strength, we have (dropping $\sim$'s)

$$\frac{\partial}{\partial r} \left(\frac{r}{R} \frac{\partial \psi}{\partial r} \right) + \frac{\partial}{\partial \theta} \left(\frac{1}{rR} \frac{\partial \psi}{\partial \theta} \right) = -r J_\phi \quad (\text{II.32'})$$

where $R = (1 + \epsilon \cos \theta) / \epsilon$, and $\epsilon = r_0 / R_0$. Expanding ψ in powers of the inverse aspect ratio and substituting into (II.32'), we get for the zero order term

$$\frac{\partial}{\partial r} \left[r \frac{\partial \psi_0}{\partial r} \right] = \begin{cases} -R_0 r J_\phi & \text{for } r < r_0, \\ 0 & \text{for } r \geq r_0, \end{cases} \quad (\text{II.33})$$

with the following conditions :

- 1) ψ_0 is finite at $r = 0$,
- 2) ψ_0 and $\partial \psi_0 / \partial r$ are continuous at $r = r_0$, and
- 3) $\psi_0 = 0$ for $r \geq r_1 > r_0$.

The last condition corresponds to placing a conducting shell at $r = r_1$. Assuming J_ϕ is uniform for $r < r_0$, a simple integration of (II.33) yields

$$\psi_0(r) = \begin{cases} -R_0 \frac{J_\phi}{4} \left[r^2 - r_0^2 \left(1 - 2 \ln \frac{r_0}{r_1} \right) \right] & r < r_0 \\ -R_0 \frac{J_\phi}{2} r_0^2 \ln \frac{r}{r_1} & r_0 \leq r < r_1 \\ 0 & r \geq r_1 \end{cases} \quad (\text{II.34})$$

Since the code is written in the cylindrical coordinate system (R, ϕ, z) , we replace r above by

$$r = [(R - R_0)^2 + z^2]^{1/2}$$

The initial poloidal field profile, and the flux contours produced by (II.34) are shown in Figs. II.5,6. Fig. II.7 shows some typical initial density and temperature profiles.

Below we summarize the typical initial conditions used in the code:

- Plasma is fully ionized and at rest, $\underline{u}=0$,
- Fill density, ρ_0 : $8 \times 10^{-9} \text{ gm/cm}^3$ (2×10^{15} of deuterium)
- Initial toroidal current, I_{ϕ_0} : $\pm 20 \text{ kA}$
- Toroidal bias field at center : 300 G
- Pinch density, ρ_p (See Fig. II.4) : $2 \rho_0$
- Background density, ρ_b : $0.4 \rho_0$
- Pinch temperature, T_p : $\leq 5 \text{ eV}$
- Background temperature, T_b : 1 eV
- Initial electrical conductivity : $1.3 \times 10^{13} / \text{sec}$ (at 1 eV)
- Thermal conductivity, K : $2 \times 10^6 \text{ erg/cm sec}^\circ \text{K}$

II.5 Solution Procedure for the Flux Function

For all the dependent variables but the flux function ψ , the equations, the initial conditions, and the boundary values discussed in the previous sections completely determine a solution in the plasma region. However, Eq. II.17, and the continuity conditions II.31 are not sufficient to determine ψ in the plasma region, since the requirement that ψ and its normal derivative be continuous on the plasma-vacuum boundary is not enough by itself to find a specific boundary value for

M = 1

TIME = 1.00E-09 SEC.

Z: BZ FIELD T: TOROIDAL FIELD (KG.) AT Z = 0. CM.

FIG. II.5. Initial field profiles

M - 1
TIME- 1.00E-09 SEC.
POLOIDAL FLUX CONTOURS

FIG. II. 6. Initial poloidal flux contours.

M - 1

TIME- 1.00E-09 SEC.

D:DENSITY +:TEMPERATURE (EV.) AT Z- 0. CM.

FIG. II.7. Initial density and temperature profiles. Density is normalized to fill density ($2 \times 10^{15}/\text{cm}^3$).

ψ . As stated earlier, the value of ψ on the wall has to come out of the solution in the infinite poloidal plane, and it is not a priori known. Therefore, we have to solve a parabolic equation for ψ (II.17) in the plasma region, and an elliptic equation (II.19) in the vacuum region and couple the two solutions through the continuity conditions (II.31). Before we present the exact mathematical formulation of the problem, we will note a feature of Tormac which somewhat simplifies the matters. Although the plasma is not enclosed in a conducting shell, there is a copper liner just inside the inner wall of the vacuum vessel (Fig. II.1) which can be used as a partial flux surface. It is not slotted and is supposed to keep the plasma off the inner wall through "flux compression". As we will see in Chapter IV, it does not really serve that purpose, but we can use it ourselves for another purpose, namely for the purpose of reducing our computational domain in the vacuum region by assuming that it extends to infinity in $\pm z$ directions. Since it is not slotted, the flux inside this liner (or the flux in the "hole")

$$\psi_h = \int_{S_{\text{hole}}} (\underline{B} \cdot \hat{z}) r dr d\phi$$

is an invariant; if it is zero initially, which we assume, then it will be zero for all times; therefore, on the copper liner the flux function ψ , which is directly proportional to the poloidal flux (See Sect. II.1) will be zero. Thus we do have a definite boundary value for ψ at least on some portion of our boundary.

Now we can present the exact mathematical problem to be solved to determine ψ in the poloidal plane:

- Solve

$$\frac{\partial \psi}{\partial t} + (\underline{u} \cdot \nabla) \psi = \frac{c^2}{4\pi\sigma} \Delta^* \psi$$

in the plasma region where

$$\Delta^* \psi = r \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \psi}{\partial r} \right) + \frac{\partial^2 \psi}{\partial z^2}$$

- Solve

$$\Delta^* \psi = - \frac{4\pi}{c} r J_{\phi e}$$

in the vacuum region.

(II.35)

- These two solutions are coupled through the relations

$$[\psi] = 0, \quad \left[\frac{\partial \psi}{\partial n} \right] = 0$$

on the non-conducting portion of the plasma-vacuum boundary.

- Moreover, ψ has to satisfy the following boundary conditions:

- $\psi = 0$ on the copper liner,

- $\lim_{r \rightarrow \infty} \psi(r, z, t) = 0, \quad \lim_{z \rightarrow \pm \infty} \psi(r, z, t) = 0,$

- $\frac{\partial \psi}{\partial z} \Big|_{z=0} = 0$, .e., we will assume symmetry about the mid-plane and work only in the upper-half plane

$$\{ (r, z), r_1 \leq r < \infty, 0 \leq z < \infty \}$$

Note that in the equations above, $\underline{u}$ is an unknown also; therefore, the whole process of solving for ψ is coupled to the solution of the equations for $\rho, \underline{u}, T$, and χ .

III. NUMERICAL METHODS

We solve Eqs. II.1-3,16, and 17 for the variables ρ , $\underline{u}$, T , ψ , and χ using finite difference methods on an Eulerian mesh. Extensive discussions of numerical methods in magnetohydrodynamic calculations can be found in the articles by Killeen [1], Roberts and Potter [2], and Brackbill [3]. The methods we use in our Tormac simulations are similar to those used by Lui and Chu (Refs. [2], and [5] in Chap. II), Sharky [4], and Park [5]. Below we will summarize our basic numerical methods and then discuss some problems peculiar to Tormac geometry.

III.1 Formulation of the Finite Difference Equations.

We use an alternating direction implicit method (ADI) [6] to solve the difference equations on a two dimensional Eulerian mesh (Fig. III.1). For numerical purposes, the hyperbolic equations for the mass continuity and momentum density (Eqs. II.1,2) are written in the form:

$$\frac{\partial \underline{W}}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (r \underline{M}) + \frac{\partial \underline{N}}{\partial z} = \underline{S} + \underline{F} \quad (\text{III.1})$$

where

$$\underline{W} = \begin{bmatrix} \rho \\ \rho u_r \\ \rho u_z \\ \rho u_\phi \end{bmatrix} \quad \underline{M} = \begin{bmatrix} \rho u_r \\ \rho u_r^2 + (\gamma-1) \rho T \\ \rho u_r u_z \\ \rho u_r u_\phi \end{bmatrix} \quad \underline{N} = \begin{bmatrix} \rho u_z \\ \rho u_r u_z \\ \rho u_z^2 + (\gamma-1) \rho T \\ \rho u_z u_\phi \end{bmatrix}$$

FIG. III.1. Eulerian mesh used in the calculations.

Typically, $N=16$, and $M=14$, which gives

$$\Delta r = \Delta z = 1 \text{ cm.}$$

$$\underline{S} = \begin{pmatrix} 0 \\ (\gamma-1) \frac{\rho T}{r} + \frac{\rho u_\phi^2}{r} \\ 0 \\ -r u_r u_\phi / r \end{pmatrix} \quad \underline{F} = \begin{pmatrix} 0 \\ J_\phi B_z - J_z B_\phi \\ J_r B_\phi - J_\phi B_r \\ J_z B_r - J_r B_z \end{pmatrix}$$

Note that the variables have been normalized as follows:

$$\begin{aligned} r &= r_d / r_0, \quad t = t_d / t_0, \quad \rho = \rho_d / \rho_0, \quad u = u_d / u_0 \\ B &= B_d / B_0, \quad J = J_d / [c B_0 / 4\pi r_0], \quad \psi = \psi_d / (r_0^2 B_0) \\ T &= c_v T_d / u_0^2, \quad \chi = \chi_d / (r_0 B_0), \quad \sigma = \sigma_d / \sigma_0, \quad K = K_d / K_0 \end{aligned}$$

where

ρ_0 : initial fill density,

B_0 : a characteristic field strength,

u_0 : a characteristic Alfven speed, $u_0 = B_0 / (4\pi\rho_0)^{1/2}$

t_0 : Alfven transit time, $t_0 = r_0 / u_0$

r_0 : the half-width of the device,

σ_0 : initial electrical conductivity,

K_0 : initial thermal conductivity,

and ϕ_d , where ϕ is any variable above, is the dimensional quantity corresponding to the non-dimensional variable ϕ used in the difference equations.

In terms of these non-dimensional variables, the parabolic equations II.3,16,17 become:

$$\frac{\partial T}{\partial t} + (\underline{u} \cdot \nabla) T = \frac{K_{00} t_0}{\rho_0 c_{v0} r_0^2} \frac{1}{\rho} \nabla \cdot [KVT] - (\gamma-1) TV \cdot \underline{u} + \frac{c^2 t_0}{4\pi r_{00}^2 \sigma_0} \frac{J^2}{\rho \sigma} \quad (\text{III.2a})$$

$$\frac{\partial \chi}{\partial t} = \frac{c^2 t_0}{4\pi r_{00}^2 \sigma_0} \left\{ r \frac{\partial}{\partial r} \left[\frac{1}{\sigma r} \frac{\partial \chi}{\partial r} \right] + \frac{\partial}{\partial z} \left[\frac{1}{\sigma} \frac{\partial \chi}{\partial z} \right] \right\} - r \frac{\partial}{\partial r} \left[\frac{u_r \chi}{r} - u_\phi B_r \right] + \frac{\partial}{\partial z} [r u_\phi B_z - u_z \chi] \quad (\text{III.2b})$$

$$\frac{\partial \psi}{\partial t} + (\underline{u} \cdot \nabla) \psi = \frac{c^2 t_0}{4\pi r_{00}^2 \sigma_0} \frac{1}{\sigma} \Delta^* \psi \quad (\text{III.2c})$$

Note that the coefficient $c^2 t_0 / 4\pi r_{00}^2 \sigma_0$ is the inverse of the magnetic Reynolds number, $S = t_r / t_0$, where

$$t_r = 4\pi \sigma_0 r_0^2 / c^2, \quad \text{the resistive diffusion time, and}$$

$$t_0 = r_0 / u_0, \quad \text{the Alfvén transit time.}$$

The solution algorithm for advancing the variables from n th to $(n+1)$ st time step, which essentially follows that of Lui and Chu (Ref. [2] in Chap. II), is as follows:

1) Calculate a predictor value $\underline{W}^{*n+1}$ using Eq. III.1. This is an explicit Lax-Friedrichs scheme that is chosen for its dissipative quality:

$$\underline{W}_{i,j}^{*n+1} = \hat{\underline{W}}_{i,j}^n - \Delta t \left[\underline{H}_r \underline{M}_{i,j}^n + \underline{H}_z \underline{N}_{i,j}^n \right] + \Delta t \left[\underline{S}_{i,j}^n + \underline{F}_{i,j}^n \right]$$

where

(III.3)

$$\hat{W}_{i,j}^n = \frac{1}{4} [\underline{W}_{i+1,j}^n + \underline{W}_{i-1,j}^n + \underline{W}_{i,j+1}^n + \underline{W}_{i,j-1}^n]$$

$$\underline{H M}_{i,j}^n = \frac{1}{2r_i \Delta r} [r_{i+1} \underline{M}_{i+1,j}^n - r_{i-1} \underline{M}_{i-1,j}^n]$$

$$\underline{H N}_{i,j}^n = \frac{1}{2\Delta z} [\underline{N}_{i,j+1}^n - \underline{N}_{i,j-1}^n]$$

and as usual $\underline{W}_{i,j}^n = \underline{W}(i\Delta r, j\Delta z, n\Delta t)$.

2) Using $\underline{W}^{n+1}$ in Eq. III.2, calculate ψ^{n+1} , χ^{n+1} , and T^{n+1} using an ADI scheme, in which the Eulerian mesh is swept alternately in r and z directions in successive time steps. χ and T calculations are straight-forward. We will have more to say about ψ calculations later in section III.4.

3) Finally calculate $\underline{W}^{n+1}$ using the newly calculated field and current values, again with an ADI scheme. This step proceeds as follows:

a) n is odd; implicit in z , explicit in r directions.

$$\begin{aligned} \frac{\underline{W}_{i,j}^{n+1} - \underline{W}_{i,j}^n}{\Delta t} + \frac{1}{2r_i \Delta r} [r_{i+1} \underline{M}_{i+1,j}^n - r_{i-1} \underline{M}_{i-1,j}^n] \\ + \frac{1}{2\Delta z} [\underline{B}_{i,j+1}^{*n+1} \cdot \underline{W}_{i,j+1}^{n+1} - \underline{B}_{i,j-1}^{*n+1} \cdot \underline{W}_{i,j-1}^{n+1}] \\ = \frac{1}{2} [\underline{S}_{i,j}^n + \underline{C}_{i,j}^{*n+1} \cdot \underline{W}_{i,j}^{n+1}] + \frac{1}{2} [\underline{F}_{i,j}^n + \underline{F}_{i,j}^{n+1}] \end{aligned} \quad (\text{III.4})$$

Note that the non-linear term $\underline{N}_{i,j}^{n+1}$ in III.1 has been linearized as

$\underline{B}_{i,j}^{*n+1} \cdot \underline{W}_{i,j}^{n+1}$, where

$$B_{i,j}^{*n+1} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & u_{z,i,j}^{*n+1} & 0 & 0 \\ (\gamma-1)T_{i,j}^n & 0 & u_{z,i,j}^{*n+1} & 0 \\ 0 & 0 & 0 & u_{z,i,j}^{*n+1} \end{pmatrix}$$

We found that writing the S term on the right hand side of III.1 as

$$\frac{1}{2} [S_{i,j}^n + S_{i,j}^{*n+1}] \quad , \text{ instead of } \frac{1}{2} [S_{i,j}^n + S_{i,j}^{*n+1}]$$

as in Lui and Chu ([5] in Chap. II), gives better numerical stability.

$C_{i,j}^{*n+1} W_{i,j}^{n+1}$ term in III.4 is the linearized form of $S_{i,j}^{*n+1}$, where

$$C_{i,j}^{*n+1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ (\gamma-1)T_{i,j}^n / r_i & 0 & 0 & u_{\phi,i,j}^{*n+1} / r_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -u_{r,i,j}^{*n+1} / r_i \end{pmatrix}$$

Eq. III.4 can be put in the form

$$-\gamma_{i,j-1} \cdot W_{i,j-1}^{n+1} + \beta_{i,j} \cdot W_{i,j}^{n+1} - \alpha_{i,j+1} \cdot W_{i,j+1}^{n+1} = D_{i,j}^{n+1} \quad (\text{III.5})$$

($j=2,3,\dots,M-1$ for each $i, i=2,3,\dots,N-1$), where

$$\gamma_{i,j-1} = \frac{\Delta t}{2\Delta z} B_{i,j-1}^{*n+1}$$

$$\beta_{i,j} = I - \frac{\Delta t}{2} C_{i,j}^{*n+1} \quad (I : \text{identity matrix})$$

$$\alpha_{i,j+1} = -\frac{\Delta t}{2\Delta z} B_{i,j+1}^{*n+1}$$

and

$$D_{i,j}^{n+1} = W_{i,j}^n - \frac{\Delta t}{2r_i \Delta r} [r_{i+1} M_{i+1,j}^n - r_{i-1} M_{i-1,j}^n] + \frac{\Delta t}{2} [S_{i,j}^n + F_{i,j}^n + F_{i,j}^{*n+1}]$$

α , β , and γ are 4×4 matrices. D is a 4×1 vector. For each $i, i=2,3,\dots,N-1$, Eq. III.5 gives a tri-diagonal block matrix of the form

[illegible]

Eq. III.6 is solved for $\underline{W}^{n+1}$ using the highly efficient Crout reduction method [7].

b) n is even; implicit in r, explicit in z directions.

$$\begin{aligned}
 \frac{\underline{W}_{i,j}^{n+1} - \underline{W}_{i,j}^n}{\Delta t} &+ \frac{1}{2r_i \Delta r} [r_{i+1} \underline{A}_{i+1,j}^{*n+1} \cdot \underline{W}_{i+1,j}^{n+1} - r_{i-1} \underline{A}_{i-1,j}^{*n+1} \cdot \underline{W}_{i-1,j}^{n+1}] \\
 &+ \frac{1}{2\Delta z} [\underline{N}_{i,j+1}^n - \underline{N}_{i,j-1}^n] \\
 &= \frac{1}{2} [\underline{S}_{i,j}^n + \underline{C}_{i,j}^{*n+1} \cdot \underline{W}_{i,j}^{n+1}] + \frac{1}{2} [\underline{F}_{i,j}^n + \underline{F}_{i,j}^{n+1}]
 \end{aligned}
 \tag{III.7}$$

which again can be written in the form

$$-\gamma_{i-1,j} \cdot \underline{W}_{i-1,j}^{n+1} + \beta_{i,j} \cdot \underline{W}_{i,j}^{n+1} - \alpha_{i+1,j} \cdot \underline{W}_{i+1,j}^{n+1} = \underline{D}_{i,j}^{n+1}
 \tag{III.8}$$

where

$$\gamma_{i-1,j} = \frac{\Delta t}{2r_i \Delta r} r_{i-1} \underline{A}_{i-1,j}^{*n+1}$$

$$\beta_{i,j} = 1 - \frac{\Delta t}{2} \underline{C}_{i,j}^{*n+1}$$

$$\alpha_{i+1,j} = -\frac{\Delta t}{2r_i \Delta r} r_{i+1} \underline{A}_{i+1,j}^{*n+1}$$

and

$$\underline{D}_{i,j}^{n+1} = \underline{W}_{i,j}^n - \frac{\Delta t}{2\Delta z} [\underline{N}_{i,j+1}^n - \underline{N}_{i,j-1}^n] + \frac{\Delta t}{2} [\underline{S}_{i,j}^n + \underline{F}_{i,j}^n + \underline{F}_{i,j}^{n+1}]$$

where

$$A^{*n+1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ (\gamma-1)T_{i,j}^n & u_{r,i,j}^{*n+1} & 0 & 0 \\ 0 & 0 & u_{r,i,j}^{*n+1} & 0 \\ 0 & 0 & 0 & u_{r,i,j}^{*n+1} \end{bmatrix}$$

and $A^{*n+1} \cdot \underline{W}^{n+1}$ is the linearized form of $\underline{M}^{n+1}$ in Eq. III.1. Eq.

III.8 produces, for each $j, j=2,3,\dots,M-1$, a tri-diagonal block matrix similar in form to Eq. III.6, which is again solved using Crout's method.

III.2 Numerical Boundary Conditions.

Boundary conditions for the partial differential equations discussed in section II.3 are usually not enough to determine the solutions of the corresponding finite difference equations. Although only one boundary condition, $u_{\perp} = 0$, is needed for the hyperbolic set III.1, one is often forced to specify extraneous boundary conditions for ρ and $u_{||}$, the components of $\underline{u}$ parallel to the wall, during numerical simulations. Here we essentially follow Ref. [2], Chap. II and set the normal derivative of these variables (actually the corresponding normal differences) equal to zero, i.e.

$$\rho_N = \rho_{N-1}, \quad u_{||N} = u_{||N-1}, \text{ etc.}$$

This boundary condition corresponds to a first order extrapolation of ρ , and $u_{||}$ at the boundary. Although it is not as accurate as methods such as integrating along the characteristics, it is known to give better results than most higher order extrapolation schemes [8], especially for long-term integrations like ours [9].

One has to be careful about how one specifies $u_{\perp} = 0$ in the difference equations. Park [5] shows that simply letting (See Fig. III.1)

$$u_{r1,j} = u_{rN,j} = 0, \quad u_{zi,M} = 0 \quad (\text{III.9})$$

may lead to considerable mass increase in the plasma region during the implosion, and that one needs to specify

$$u_{r1,j} = -u_{r2,j}, \quad u_{rN,j} = -u_{rN-1,j}, \quad u_{zi,M} = -u_{zi,M-1} \quad (\text{III.10})$$

to ensure zero mass flux through the boundary. We follow his suggestion and adopt III.10 during that portion of the simulations when the plasma is imploding, but we find that during the expansion phase when the plasma is flowing towards the walls, III. 10 leads to numerical oscillations of large amplitude. These oscillations stay at "harmless" levels if we use III.9 during expansion. This is probably because of the fact that with III.10, the walls become a better reflector than with III.9. In the code, we switch back and forth between III.9 and III.10 as follows:

$$u_{n_w} = \begin{cases} -u_{n_{w-1}} & \text{if } \hat{n} \cdot \underline{u}_{w-1} \leq 0 \\ 0 & \text{if } \hat{n} \cdot \underline{u}_{w-1} > 0 \end{cases} \quad (\text{III.11})$$

where $\hat{n}$ is the outward normal to the wall, $u_{n_w} = u_{\perp}$ at the wall, and $u_{n_{w-1}} = u_{\perp}$ one grid point inside the wall.

The parabolic equations III.2 do not need extraneous boundary conditions. As described in section II.3, we let

$$\chi_w(t) = \frac{2I_z(t)}{c}$$

where I_z is the total current through the "hole". We keep the walls at a constant temperature, $T_w = 1$ eV.

The implementation of the continuity conditions, $[\psi] = [\partial\psi / \partial n] = 0$, will be discussed later in section III.4b.

As mentioned earlier, we assume symmetry about the mid-plane, $z = 0$. By inspection of our equations and from physical considerations we see that u_z , u_ϕ , J_r , and B_r have odd symmetry about $z = 0$, whereas ρ , u_r , T , ψ , χ , J_z , J_ϕ , B_ϕ , B_z , K , σ have even symmetry. In the calculations, $J=2$, $I=1,2,\dots,N$ (See Fig. III.1) corresponds to the mid-plane about which the variables are reflected as even or odd, depending on their nature of symmetry, e.g.

$$u_{z,i,1} = -u_{z,i,3}, \text{ but } \rho_{i,1} = +\rho_{i,3}, \text{ etc.}$$

III.3 Low Density Region and Artificial Viscosity

As in Lui and Chu ([2] in Chap. II), we do not allow a vacuum to form behind the imploding plasma in our "plasma region". Instead we keep a floor value of density, ρ_m , below which we do not allow it to fall. This method of allowing a low density plasma region in which there may be currents flowing, used essentially for numerical reasons, is probably more realistic than assuming the creation of a plasma and

current-free vacuum region behind the imploding plasma, since 100% sweeping efficiency by magnetic fields is unattainable. Low densities, which in general occur in the high field regions, imply high Alfvén speeds. Numerical stability and accuracy requirements, for a given minimum step size Δt , put an upper limit on how large an Alfvén speed one can allow in the calculations, which in turn puts a lower bound on ρ_m . In the code ρ_m is typically 10% of the initial fill density.

The ADI scheme we use, when applied to a pure initial value problem with constant coefficients, gives unconditional numerical stability [6], i.e., stability independent of Δt , Δr , or Δz . But unfortunately in a "real-life situation" such as ours, where the equations are non-linear and boundaries play an important role, there is a limit on how large Δt can be made for a fixed mesh size before one gets into numerical difficulties. These difficulties usually manifest themselves as oscillations in the values of the variables, the amplitudes of which usually saturate at some intolerably high level, or sometimes increase exponentially, leading to the breakdown of the calculations. These oscillations, generally attributed to "non-linear" numerical instabilities, limit the size of Δt well below that allowed by a linear stability analysis of the difference equations. In our calculations the Courant number [6] L_f is typically much less than unity, where

$$L_f = \{u^2 + c^2\}^{1/2} \frac{\Delta t}{\Delta x}$$

Above u is the fluid velocity, and c is a characteristic signal speed, e.g., magnetosonic speed in the plasma.

To smooth out the sharp gradients in the calculations, which are believed to be at least partially responsible for the non-linear numerical instabilities, we introduce an artificial viscosity term of the Lapidus type [10] to the fluid equations. This dissipative term, which is added to the right hand side of Eq. III.1, is of the form

$$D = a(\Delta r)^2 \frac{\partial}{\partial r} \left\{ \left| \frac{\partial \underline{u}}{\partial r} \right| \frac{\partial \underline{W}}{\partial r} \right\} + a(\Delta z)^2 \frac{\partial}{\partial z} \left\{ \left| \frac{\partial \underline{u}}{\partial z} \right| \frac{\partial \underline{W}}{\partial z} \right\}$$

where a is some constant less than 1, and $\underline{W}$ is the vector $(\rho, \rho u_r, \rho u_z, \rho u_\phi)$. Note that the effect of this term is to smooth $\underline{W}$ in the regions where the gradients are large; it leaves $\underline{W}$ essentially unchanged in the rest of the computational domain.

III.4 Solution for the Poloidal Flux Function

The rest of this chapter will be devoted to the discussion of the analytic and numerical methods used in the solution of the Eq. II.35 for the poloidal flux function ψ . As mentioned earlier solution of II.35 entails solving an elliptic equation in an infinite domain outside the plasma region. Of course for numerical purposes, working with an infinite mesh placed over an infinite domain is out of the question; therefore, we either reduce our computational domain to a manageable finite size, or reformulate the equations in such a form as not to involve infinite domains.

In the simplest method, one effectively "redefines infinity" by imposing the boundary condition at infinity, $\psi = 0$, only a finite

distance away from the plasma region (Fig. III.2). Physically this corresponds to enclosing the experimental set up in a virtual, perfectly conducting shell that is used as a flux surface in the computations. Of course the external currents I_1 , I_2 , and I_3 induce surface currents on this shell, and one tries to minimize their effects on the field profiles by putting the shell at a distance sufficiently far away from the plasma-vacuum boundary. The farther away it is, the more accurate the results are. Unfortunately obtaining sufficient accuracy is expensive in terms of computer time, since the area over which the elliptic vacuum equation has to be solved increases quadratically with the distance between the plasma-vacuum boundary and the virtual conducting shell.

One can improve upon this simple method by subtracting off the contributions to ψ from the surface currents on the virtual conductor using Green's function techniques. In this method, one not only improves the accuracy of the solution by getting rid of non-physical surface currents, but presumably also decreases the computation time by reducing the area over which the vacuum equation is solved, since there is no longer a need to place the virtual shell far away from the plasma region. This method is used in ideal MHD equilibrium calculations of systems without conducting shells [11],[12] and proceeds as follows (Fig. III.3): First one solves

$$\begin{aligned} \Delta^* \tilde{\psi} &= -g(r,z) \quad \text{in } R \text{ (plasma region),} \\ \Delta^* \tilde{\psi} &= 0 \quad \text{in } R^* \text{ (vacuum region),} \end{aligned} \quad (\text{III.12})$$

with the following boundary conditions:

FIG. III.2. Computational domain enclosed by a virtual conducting shell (See Sect. III.4).

FIG. III.3. An arbitrary computational domain and the boundaries used in an ideal MHD equilibrium calculation with Green's function technique.

$\tilde{\psi} = 0$ on ∂R^* (virtual conducting shell), and $\tilde{\psi}$ and $\partial\tilde{\psi}/\partial n$ are continuous on ∂R , the plasma vacuum boundary. $\tilde{\psi}$ can be written as

$$\tilde{\psi} = \psi + \psi_S$$

where ψ is the solution of the infinite domain problem, and ψ_S is the contributions from the surface currents on ∂R^* . Since $\tilde{\psi} = 0$ outside of ∂R^* , the surface current $J_S(s^*)$ is simply given by

$$J_S(s^*) = \frac{c}{4\pi} \frac{1}{r^*} \frac{\partial\tilde{\psi}(s^*)}{\partial n}$$

where $\hat{n}$ is the outward normal to ∂R^* . Using the known Green's function $G(\underline{x}, \underline{x}')$ [11], for the infinite domain problem,

$$G(\underline{x}, \underline{x}') = \frac{1}{\pi} \frac{\sqrt{\underline{x} \cdot \underline{x}'}}{k} \left\{ \left(1 - \frac{k^2}{2}\right) K(k) - E(k) \right\}, \text{ where} \quad (\text{III.13})$$

$$k^2 = \frac{4rr'}{(r+r')^2 + (z-z')^2}, \quad \underline{x} = (\hat{r})r + (\hat{z})z,$$

and $K(k)$, and $E(k)$ are complete elliptic integrals of the first, and the second kind respectively, ψ_S can be written as

$$\psi_S(\underline{x}) = \oint_{\partial R^*} \frac{1}{r'} G(\underline{x}, \underline{x}') \frac{\partial\tilde{\psi}(\underline{x}')}{\partial n'} ds' \quad (\text{III.14})$$

Hence the solution to the infinite domain problem is

$$\psi(\underline{x}) = \tilde{\psi}(\underline{x}) - \psi_S(\underline{x})$$

where $\tilde{\psi}$ solves III.12, and ψ_S is given by III.14. Note that

$$\Delta^* \psi_S = -\frac{4\pi}{c} r J_S(\underline{x}) \delta(\underline{x} - \underline{x}^*), \quad \underline{x}^* \in \partial R^*.$$

Thus

$$\Delta^* \psi = \Delta^* \tilde{\psi} - \Delta^* \psi_s = \begin{cases} -g(r,z) & \text{in } R, \\ 0 & \text{in } R^*, \\ 0 & \text{on } \partial R^*, \\ 0 & \text{outside of } \partial R^*, \end{cases}$$

and $\psi \rightarrow 0$ as $|x| \rightarrow \infty$, since $\tilde{\psi}$ is zero outside of ∂R^* , and

$$\lim_{|x| \rightarrow \infty} \psi_s(x) = 0.$$

Therefore, ψ is the solution to the infinite domain problem.

Using this method in our initial-boundary value problem proves to be more complicated than one would expect. To study the feasibility of this procedure, we applied it to a simplified Tormac-like geometry in which the plasma is replaced by a solid conductor of finite conductivity [13]. (Note that we will continue to refer to this conductor as the "plasma region".) In this case the equation in the plasma region of Fig. III.3 becomes

$$\frac{\partial \psi}{\partial t} = \eta \Delta^* \psi, \quad \eta = \frac{c^2}{4\pi\sigma}$$

Because of the external currents, the vacuum equation is now

$$\Delta^* \psi = -\frac{4\pi}{c} r J_e$$

Proceeding as before, we first solve

$$\begin{aligned}\frac{\partial \tilde{\psi}}{\partial t} &= \eta \Delta^* \tilde{\psi} & \text{in } R, \\ \Delta^* \tilde{\psi} &= -\frac{4\pi}{c} r J_e & \text{in } R^*,\end{aligned}\quad (\text{III.15})$$

with $\tilde{\psi} = 0$ on ∂R^* , and the usual continuity conditions on ∂R . We then calculate the contributions from the surface currents on ∂R^* using III.14. Again we let $\psi = \tilde{\psi} - \psi_s$ and check to see if ψ solves the infinite domain problem:

a) In the vacuum region,

$$\begin{aligned}\Delta^* \psi &= \Delta^* \tilde{\psi} - \Delta^* \psi_s = -\frac{4\pi}{c} r J_e & \text{in } R^*, \\ \Delta^* \psi &= 0 & \text{on } \partial R^* \text{ and elsewhere where } J_e = 0,\end{aligned}$$

and $\psi \rightarrow 0$ as $|\underline{x}| \rightarrow \infty$. Thus, ψ satisfies the vacuum region equations and boundary conditions.

b) In the plasma region,

$$\frac{\partial \psi}{\partial t} = \frac{\partial \tilde{\psi}}{\partial t} - \frac{\partial \psi_s}{\partial t} = \eta \Delta^* \tilde{\psi} - \frac{\partial \psi_s}{\partial t} \quad (\text{III.16})$$

which can be written as

$$\frac{\partial \psi}{\partial t} = \eta \Delta^* \psi - \frac{\partial \psi_s}{\partial t}$$

since $\Delta^* \psi_s = 0$ in R . Recall that

$$\psi_s = \psi_s(\underline{x}, t) = \oint_{\partial R^*} \frac{1}{r'} G(\underline{x}, \underline{x}') \frac{\partial \tilde{\psi}(\underline{x}', t)}{\partial n'} ds'$$

Thus, $\partial \psi_s / \partial t \neq 0$, and ψ does not satisfy the equations in the plasma region, i.e., $\psi = \tilde{\psi} - \psi_s$ is not the solution to the infinite domain problem! Physically, what is happening is that when we "turn off" the surface currents on ∂R^* by subtracting off their contributions to $\tilde{\psi}$, the plasma responds by changing the plasma current J_p ; it has no way of distinguishing between the external

currents on the toroidal coils and the surface currents on the virtual conducting shell, and it will react to changes in either one. In fact, the term $\partial\psi_s/\partial t$ in III.16 is related to the change in the plasma current as follows:

$$\delta J_p(\underline{x}, t) = - \frac{\sigma}{c} \frac{1}{r} \frac{\partial \psi_s(\underline{x}, t)}{\partial t} \quad (\text{III.17})$$

In summary, $\tilde{\psi}$ of Eq. III.15 is determined by the plasma current J_p , the external currents on the coils J_e , and the surface currents J_s on ∂R^* . If J_p stayed unaffected when we turn off J_s , then $\psi = \tilde{\psi} - \psi_s$ would be the correct solution. But unfortunately, the plasma and the virtual conducting shell are inductively coupled[†], and letting $J_s \rightarrow 0$ induces additional currents in the plasma and changes J_p . As a result $\psi = \tilde{\psi} - \psi_s$, being based on the old value of the plasma current, is no longer the correct answer, unlike in the case of the equilibrium calculations.

Then how does one calculate the correct flux function ψ for the infinite domain problem using Green's function techniques? At this point one way to proceed is to use an iterative method. We know that $\psi = \tilde{\psi} - \psi_s$ is incorrect in the plasma region, but if we just calculate a new boundary value on ∂R^* (instead of $\tilde{\psi} = 0$)

$$\tilde{\psi}^{k+1}(s^*) = \tilde{\psi}^k(s^*) - \psi_s^k(s^*)$$

† There is obviously a similar coupling between the plasma current and the external currents in the coils. Although the equations properly take into account the effects of J_e on J_p , the reverse is not true, since we do not have circuit equations for the external currents but "force" them to have specific values.

where k is an iteration index, and calculate $\tilde{\psi}^{k+1}$ for all x in R and R^* using this new boundary value, then we will have taken into account the changes in the plasma current induced by the "turning off" of surface currents on ∂R^* . But now if return to the boundary ∂R^* , we will see that there are new surface currents on it, although hopefully of smaller magnitude, which have been induced by the changes in the plasma current. When we turn off these new currents by letting

$$\tilde{\psi}^{k+2}(s^*) = \tilde{\psi}^{k+1}(s^*) - \psi_s^{k+1}(s^*)$$

again the plasma current will change, which in turn will induce new surface currents, and so on. Although we have no formal proof, one would expect, on physical grounds, $\tilde{\psi}^k$ to eventually converge to the exact solution ψ of the infinite domain problem.

A code was written using the algorithm described above to study the convergence characteristics of this procedure. The algorithm proceeds as follows:

i) At the end of the n th time step, ψ^n is known for all the points in R and R^* (Fig. III.3), and there are no surface currents. We increase the external currents to their new values, and as an initial guess to the new boundary value on ∂R^* we use

$$\tilde{\psi}^1(s^*) = \psi^n(s^*)$$

ii) Calculate $\tilde{\psi}^k$, which we hope will eventually converge to ψ^{n+1} , using

$$\frac{\partial \tilde{\psi}^k}{\partial t} = \eta \Delta^* \tilde{\psi}^k \quad \text{in the plasma region } R,$$

$$\Delta^* \tilde{\psi}^k = -\frac{4\pi}{c} r J_e^{n+1} \quad \text{in the vacuum region } R^*,$$

with $\tilde{\psi}^k = \tilde{\psi}^k(s^*)$ on ∂R^* , and $[\tilde{\psi}] = [\partial \tilde{\psi} / \partial n] = 0$ on ∂R . If

$$\max_{i,j} \left| \frac{\tilde{\psi}_{i,j}^k - \tilde{\psi}_{i,j}^{k-1}}{\tilde{\psi}_{i,j}^{k+1}} \right| < \varepsilon$$

where $\varepsilon = 10^{-3}$, we assume the iterations have converged and let

$$\psi_{i,j}^{n+1} = \tilde{\psi}_{i,j}^k$$

iii) Otherwise, calculate the new surface currents $J_s^k(s^*)$ on ∂R^* .

Using the boundary condition

$$\hat{n} \times [\underline{B}] = \frac{4\pi}{c} \underline{J}_s$$

we have

$$\underline{J}_s^k = (-\hat{\phi}) \frac{c}{4\pi} \frac{1}{r} \left[\frac{\partial \tilde{\psi}^k}{\partial n} \right] \quad (\text{III.18})$$

where (See Fig. III.4)

$$\left[\frac{\partial \tilde{\psi}}{\partial n} \right] = \left. \frac{\partial \tilde{\psi}(s^*)}{\partial n} \right|_2 - \left. \frac{\partial \tilde{\psi}(s^*)}{\partial n} \right|_1$$

Note that to be able to calculate $\partial \tilde{\psi} / \partial n|_2$ on ∂R^* , we have to know the values of $\tilde{\psi}$ at least on one column of mesh points parallel to ∂R^* , outside of ∂R^* , for an accuracy of order Δx ; we would need two columns for second order accuracy. For instance,

$$J_{s,i,j}^k = -\frac{c}{4\pi} \frac{1}{r_w} \left\{ \frac{\tilde{\psi}_{w+1,j}^k - 2\tilde{\psi}_{w,j}^k + \tilde{\psi}_{w-1,j}^k}{\Delta x} \right\} + O(\Delta x) \quad (\text{III.18'})$$

FIG. III.4. Computational mesh around the virtual conducting shell ∂R^* . One or more rows of points outside of ∂R^* (in region 2) are needed to calculate the surface currents on ∂R^* .

is only first order accurate, but we need to know only $\tilde{\psi}_{w+1,j}$ outside of our usual computational domain (1) in Fig. III.4). At the end of each iteration $\tilde{\psi}_{w+1,j}$ is updated as follows:

$$\tilde{\psi}_{w+1,j}^{k+1} = \tilde{\psi}_{w+1,j}^k - \psi_{s_{w+1,j}}^k$$

where

$$\psi_{s_{w+1,j}}^k = \frac{4\pi}{c} \oint_{\partial R^*} G(\underline{x}, s') J_s^k(s') ds' \quad (\text{III.19})$$

$$\underline{x} = \hat{r} (w+1) \Delta r + \hat{z} j \Delta z,$$

and J_s^k is given by Eq. III.18'. Note that $\psi_{s_{w+1,j}}^k$ is the contribution from the surface currents on ∂R^* to $\tilde{\psi}$ at the point $((w+1) \Delta r, j \Delta z)$.

iv) Update the boundary values on ∂R^* :

$$\tilde{\psi}_{w,j}^{k+1} = \tilde{\psi}_{w,j}^k - \psi_{s_{w,j}}^k$$

where

$$\psi_{s_{w,j}}^k = \frac{4\pi}{c} \oint_{\partial R^*} G(\underline{x}, s') \langle J_s^k(s') \rangle ds'$$

with $\underline{x} = \hat{r} (w \Delta r) + \hat{z} j \Delta z$. Note that during this integration, any singularities in the integrand are avoided by defining ψ_s at the mesh points ($\underline{x}$'s in Fig. III.4), and the currents $\langle J_s^k \rangle$ on the circles between the mesh points. $\langle J_s^k \rangle$ is actually an average value defined as

$$\langle J_s^k(s^*) \rangle \equiv J_{s_{w,j+1/2}}^k = (J_{s_{w,j}}^k + J_{s_{w,j+1}}^k) / 2 + O(\Delta x^2)$$

At this point we return to step (ii), and the whole process is repeated until convergence.

Although the iterations do converge (Figs. III.5,6), this method is computationally expensive. The solution for $\tilde{\psi}$ (step (ii)), which has to be repeated many times at each time step, is itself an iterative and time consuming step. The two line integrations for $\tilde{\psi}_S^k(w,j)$ and $\tilde{\psi}_S^k(w+1,j)$, and the storage of the Green's function $G(x,x')^\dagger$, which requires $2N^2$ memory locations (N is the total number of mesh point along the boundary ∂R^*), also contribute to the high cost of this method. Since the flux function has to be evaluated at each time step, for many thousand time steps, we were forced to look for a non-iterative method in which the step (ii) above, or its equivalent, needs to be executed only once for each time step, i.e., a method where we do not have to iterate for the boundary values on some imaginary boundary ∂R^* . One such method will be discussed below.

III.4a Mapping of the Infinite Domain to a Finite Domain

In the previous section we reduced the problem of calculating ψ on an infinite domain to a finite-domain problem by artificially placing a boundary a finite distance away from the plasma region and later subtracting off the undesirable effects introduced by this artificial boundary.

$^\dagger G(x,x')$ is calculated only once at the beginning of the calculations.

FIG. III.5. Flux function ψ found using Green's function technique, at the end of first iteration at $t = 0.02$ (normalized to diffusion time). Because of the surface currents, the "open" field lines from the previous time step are tied to the virtual conductor, and the newly formed field lines are closed and tangent to it.

FFME 10

FIG. III. 6. Flux contours at $t = 0.02$ after five iterations, with no surface currents left on the virtual conducting shell. Some of the field lines have opened, letting flux out of the artificial boundary ∂R^* (Compare with the previous figure).

An altogether different approach is to map the infinite physical domain to a finite domain through a coordinate transformation. We solve the appropriate equations for ψ on this new finite, but non-physical domain and then transform back to our poloidal plane. A set of coordinate transformations we may use is

$$M: \quad \rho = \frac{1}{r+a}, \quad \zeta = \frac{1}{z+b} \quad (\text{III.20})$$

where a and b are some positive definite constants. It is easily seen that with this transformation, the upper half of the poloidal plane

$$P : \{(r, z), 0 \leq r < \infty, 0 \leq z < \infty\}$$

is mapped onto

$$P' : \{(\rho, \zeta), 0 \leq \rho \leq \frac{1}{a}, 0 \leq \zeta \leq \frac{1}{b}\}.$$

What happens to our computational domain is shown in Fig. III.7. In the (ρ, ζ) coordinates the vacuum equation

$$\Delta^* \psi = r \frac{\partial}{\partial r} \left\{ \frac{1}{r} \frac{\partial \psi}{\partial r} \right\} + \frac{\partial^2 \psi}{\partial z^2} = -r J_e(r, z)$$

becomes

$$\rho(1-a\rho) \frac{\partial}{\partial \rho} \left\{ \frac{\rho^3}{1-a\rho} \frac{\partial \tilde{\psi}}{\partial \rho} \right\} + \zeta^2 \frac{\partial}{\partial \zeta} \left\{ \zeta^2 \frac{\partial \tilde{\psi}}{\partial \zeta} \right\} = - \left\{ \frac{1-a\rho}{\rho} \right\} \tilde{J}_e(\rho, \zeta) \quad (\text{III.21})$$

where

$$\tilde{\psi}(\rho, \zeta) = \psi\{\rho(r), \zeta(z)\}$$

and

$$\tilde{J}_e(\rho, \zeta) = J_e\{\rho(r), \zeta(z)\}$$

The exact boundary conditions given in Eq. II.35 (Chap. II) now become:

FIG. III.7. With the coordinate transformation M, the infinite upper half plane to the right of the copper liner in (a) is mapped onto the bounded domain in (b).

- 1) $\tilde{\psi}(\rho, \zeta) = 0$ for $\rho = 0, 0 \leq \zeta \leq \frac{1}{b}$
- $\tilde{\psi}(\rho, \zeta) = 0$ for $\zeta = 0, 0 \leq \rho \leq 1 / (r_1 + a)$
- 2) $\tilde{\psi}(\rho, \zeta) = 0$ for $\rho = 1/(r_1 + a), 0 \leq \zeta \leq 1/b$ (III.22)
- 3) $\partial\tilde{\psi}/\partial\zeta = 0$ for $\zeta = 1/b, 0 \leq \rho \leq 1 / (r_1 + a)$
- 4) On the plasma-vacuum boundary ∂R :

$$\tilde{\psi} \{ \rho(r), \zeta(z) \} = \psi(r, z)$$

where $\psi(r, z)$ is the solution of the usual parabolic equation in the plasma region, and

$$-\rho^2 \frac{\partial \tilde{\psi} \{ \rho(r), \zeta(z) \}}{\partial \rho} = \frac{\partial \psi(r, z)}{\partial r}, \text{ for } \rho = 1/(r_2 + a), \frac{1}{z_1 + b} \leq \zeta \leq \frac{1}{b}$$

and (III.23)

$$-\zeta^2 \frac{\partial \tilde{\psi} \{ \rho(r), \zeta(z) \}}{\partial \zeta} = \frac{\partial \psi(r, z)}{\partial z}, \text{ for } \zeta = \frac{1}{z_1 + b}, \frac{1}{r_2 + a} \leq \rho \leq \frac{1}{r_1 + a}$$

Above (1) corresponds to the boundary condition $\psi = 0$ at infinity. (2) is the similar condition on the copper liner. (3) is a statement of the assumed symmetry about the mid-plane, which is at $\zeta = 1/b$ now. The conditions in (4) are the usual continuity relations for ψ and $\partial\psi/\partial n$ on the plasma-vacuum boundary.

Our computational domain is now :

$$P' : \{ (\rho, \zeta), 0 \leq \rho \leq \frac{1}{r_1 + a}, 0 \leq \zeta \leq \frac{1}{b} \}.$$

Before discussing the solution procedure in the (ρ, ζ) coordinates, we would like to make a few comments about the apparent singularities in the coefficients in Eq. III.21. $\rho = 0$, $\rho = 1/a$, and $\zeta = 0$ are singular "lines" of III.21, and as it is well known, we can not assign arbitrary boundary conditions at singularities; the

values of $\tilde{\psi}$ at these locations have to come out of the solution of III.21 itself. $\rho = 1/a$ corresponds to $r = 0$ which itself is a singular line for the equation in (r, z) coordinates; thus, we seem to have "conserved" a singularity in going from (r, z) to (ρ, ζ) coordinates. Since $\rho = 1/a$ is in our case out of our computational domain, we need not be concerned with it anymore. But the singularities at $\rho = 0$, and $\zeta = 0$ do concern us. Although analytically III.21 does not require boundary conditions on either the ρ or ζ axis, except for the requirement that $\tilde{\psi}(\rho, \zeta)$ be finite and continuous there, during the numerical solution of III.21 we will have to specify extraneous boundary conditions on these lines, namely $\tilde{\psi} = 0$, and we have to make sure that the values we specify are compatible with the solution of the equation. To answer the question of compatibility, we have to check if $\tilde{\psi} = 0$ is the correct solution on the coordinate axes of the problem as defined above.

First, let $\rho = 0$ in III.21. Note that since there are no currents at infinity, the right hand side of III.21 for $\rho = 0$ (or $\zeta = 0$) is identically zero. Thus, we have

$$\zeta^2 \frac{\partial}{\partial \zeta} \left\{ \zeta^2 \frac{\partial \tilde{\psi}}{\partial \zeta} \right\} = 0,$$

or

$$\tilde{\psi}(\rho=0, \zeta) = C_1/\zeta + C_2$$

where C_1 and C_2 are some constants. Since $\tilde{\psi}$ is finite everywhere, we must have $C_1 = 0$; therefore,

$$\tilde{\psi}(\rho=0, \zeta) = C_2$$

A similar argument for $\zeta = 0$ leads to

$$\tilde{\psi}(\rho, \zeta = 0) = C_3$$

where C_3 is another constant. Since $\tilde{\psi}$ is assumed to be a continuous function, at $\rho = 0, \zeta = 0$, we have

$$\tilde{\psi}(0, 0) = C_2 = C_3 = C$$

Thus, on both axes $\tilde{\psi}$ is equal to the same constant C . But on the copper liner, which cuts the ρ -axis at $\rho = 1/(r_1 + a)$, we have $\tilde{\psi} = 0$. Thus, again using the continuity argument, we see that the constant C is equal to zero, i.e., $\tilde{\psi} = 0$ is the correct solution on the coordinate axes, and the extraneous boundary conditions we will use for the difference equations are compatible with the exact solution and will not pose any problems.

It is possible to transform all our equations, Eqs. III.1,2, to (ρ, ζ) coordinates and work solely in this coordinate system. But as we see in going from

$$\Delta^* \psi = -r J_{\phi e}$$

to III.21, the equations tend to be messier in the (ρ, ζ) coordinates than in the usual (r, z) coordinates. Since the numerical solution of III.1 and 2 poses enough difficulties as they are, we will not complicate the problem anymore than we have to; we will solve only the vacuum equation for the flux function ψ in this new system, and solve the

rest of the equations in the (r,z) coordinates. The actual numerical solution of the vacuum and the plasma equations for ψ is discussed in the following sections.

III.4b The Solution Algorithm and the Difference Equations

As mentioned in section III.4 during the discussion of the Green's function technique, the solution of the parabolic equation in the plasma region and the elliptic equation in the vacuum region, with given boundary conditions, is itself an iterative procedure. In the Green's function method, these "inner" iterations were repeated at each step of the "outer" iterations for the exact boundary values on ∂R^* . With the mapping technique, there is no longer a need for the outer iterations, since we are able to use the known exact boundary values, but we still have to use the inner iterations to solve the coupled elliptic-parabolic system for the flux function ψ . The algorithm for these iterations is as follows:

Let ∂R be the plasma-vacuum boundary (See Fig. III.7), and ψ_b be the value of ψ on ∂R . Let ψ^n and $\tilde{\psi}^n$ be the known values of the flux function in the plasma and vacuum regions, respectively, at the end of the n th time step, and let $\underline{W}^{*n+1}$ be the "predicted" value for the vector $\underline{W} = (\rho, \rho u_r, \rho u_z, \rho u_\phi)$ calculated using Eq. III.3. Then

- i) Solve

$$\tilde{\tilde{\Delta}}\psi^{k+1} = -\left\{\frac{1-a\rho}{\rho}\right\} \tilde{j}_e^{n+1}$$

in the vacuum region using the boundary values given in III.22 on the "outer" boundary (Fig. III.7), and $\tilde{\psi}_b^{k+1}$ on the plasma-vacuum boundary ∂R . The operator $\tilde{\Delta}$ is defined as (See Eq. III.21)

$$\tilde{\Delta} = \rho(1-a)\rho \frac{\partial}{\partial \rho} \left\{ \frac{\rho^3}{1-a\rho} \frac{\partial}{\partial \rho} \right\} + \zeta^2 \frac{\partial}{\partial \zeta} \left\{ \zeta^2 \frac{\partial}{\partial \zeta} \right\}$$

Note that k is the iteration index in the $(n+1)$ st time step.

ii) Calculate the new boundary values $\tilde{\psi}_b^{*k+1}$ using the continuity relations III.23. The third order accurate equations used in this step are (See Appendix I for the derivations):

a) On the vertical boundary, $\rho = 1/(r_2 + a), 1/(z_1 + b) \leq \zeta \leq 1/b,$

$$\begin{aligned} \tilde{\psi}_{b,i,j}^{*k+1} = & -\{a_i \Delta \rho_i [\tilde{\psi}_{i-2,j}^{k+1} - b_i^2 \tilde{\psi}_{i-1,j}^{k+1}] / (b_i - 1) \\ & + b_i \Delta \rho_{i-1} [\tilde{\psi}_{i+2,j}^k - a_i^2 \tilde{\psi}_{i+1,j}^k] / (a_i - 1)\} \\ & / \{a_i (b_i + 1) \Delta \rho_i + b_i (a_i + 1) \Delta \rho_{i-1}\} \end{aligned} \quad (\text{III.24})$$

where

$$a_i = (\rho_{i+2} - \rho_i) / (\rho_{i+1} - \rho_i)$$

$$b_i = (\rho_i - \rho_{i-2}) / (\rho_i - \rho_{i-1})$$

$$\Delta \rho_i = \rho_{i+1} - \rho_i$$

b) On the horizontal boundary, $\zeta = 1/(z_1 + b), 1/(r_2 + a) \leq \rho \leq 1/(r_1 + a)$

$$\begin{aligned}
 \tilde{\psi}_{b,i,j}^{*k+1} = & -\{a_j \Delta \zeta_j [\tilde{\psi}_{i,j-2}^{k+1} - b_j^2 \tilde{\psi}_{i,j-1}^{k+1}] / (b_j - 1) \\
 & + b_j \Delta \zeta_{j-1} [\tilde{\psi}_{i,j+2}^k - a_j^2 \tilde{\psi}_{i,j+1}^k] / (a_j - 1)\} \\
 & / \{a_j (b_j + 1) \Delta \zeta_j + b_j (a_j + 1) \Delta \zeta_{j-1}\}
 \end{aligned} \tag{III.25}$$

where

$$a_j = (\zeta_{j+2} - \zeta_j) / (\zeta_{j+1} - \zeta_j)$$

$$b_j = (\zeta_j - \zeta_{j-2}) / (\zeta_j - \zeta_{j-1})$$

$$\Delta \zeta_j = \zeta_{j+1} - \zeta_j$$

iii) Solve

$$\frac{\partial \psi}{\partial t}^{k+1} + (\underline{u}^{*n+1} \cdot \nabla) \psi^{k+1} = \eta \Delta^* \psi^{k+1}$$

in the plasma region using (r, z) coordinate system, and the boundary values

$$\psi_b^{k+1} = \tilde{\psi}_b^{*k+1}$$

on ∂R .

iv) Calculate the boundary values ψ_b^{k+2} on ∂R , using III.24,25, where now $\tilde{\psi}_{i+1,j}^k$ etc., are replaced by the newly calculated $\tilde{\psi}_{i+1,j}^{k+1}$, etc.

Stop if

$$\max_{i,j} \left| \frac{\psi_{i,j}^{k+1} - \psi_{i,j}^k}{\psi_{i,j}^{k+1}} \right| < \epsilon \quad \epsilon = 10^{-4}$$

and let $\psi_{i,j}^{n+1} = \psi_{i,j}^{k+1}$; otherwise go back to step (i) again, with $k+1$ replaced by $k+2$, etc.

In step (iii) a usual ADI method is used in which the 2-D mesh is swept in r-direction if n (not k) is odd, and in z-direction if it is even. The difference equations in this step can be written in the form :

a) n is odd; implicit in z direction,

$$-C_{i,j} \psi_{i,j-1}^{n+1} + B_{i,j} \psi_{i,j}^{n+1} - A_{i,j} \psi_{i,j+1}^{n+1} = D_{i,j}^n \quad (\text{III.26a})$$

where

$$\begin{aligned} C_{i,j} &= \frac{\Delta t}{S} \frac{1}{\sigma_{i,j}^n} \frac{1}{(\Delta z)^2} + \frac{u_{z,i,j}^{*n+1} \Delta t}{2\Delta z}, \\ B_{i,j} &= 1 + \frac{\Delta t}{S} \frac{1}{\sigma_{i,j}^n} \frac{2}{(\Delta z)^2}, \\ A_{i,j} &= \frac{\Delta t}{S} \frac{1}{\sigma_{i,j}^n} \frac{1}{(\Delta z)^2} - \frac{u_{z,i,j}^{*n+1} \Delta t}{2\Delta z}, \\ D_{i,j}^n &= \psi_{i,j}^n + \Delta t / (S \sigma_{i,j}^n) \left\{ \frac{r_i}{(\Delta r)^2} \left[\frac{\psi_{i+1,j}^n - \psi_{i,j}^n}{r_{i+1/2}} - \frac{\psi_{i,j}^n - \psi_{i-1,j}^n}{r_{i-1/2}} \right] \right. \\ &\quad \left. - u_r^n \Delta t [\psi_{i+1,j}^n - \psi_{i-1,j}^n] / 2\Delta r \right\} \end{aligned}$$

$$S = \frac{4\pi\sigma_0^2 / c^2}{r_0 / u_0}, \quad \text{characteristic magnetic Reynolds number, and}$$

b) n is even; implicit in r-direction,

$$-C_{i,j} \psi_{i-1,j}^{n+1} + B_{i,j} \psi_{i,j}^{n+1} - A_{i,j} \psi_{i+1,j}^{n+1} = D_{i,j}^n \quad (\text{III.26b})$$

where

$$\begin{aligned} C_{i,j} &= \frac{\Delta t}{S} \frac{1}{\sigma_{i,j}^n} \frac{r_i}{(\Delta r)^2} \frac{1}{r_{i-1/2}} + u_{r,i,j}^{*n+1} \Delta t / 2\Delta r \\ B_{i,j} &= 1 + \frac{\Delta t}{S} \frac{1}{\sigma_{i,j}^n} \left\{ \frac{r_i}{(\Delta r)^2} \left[\frac{1}{r_{i+1/2}} + \frac{1}{r_{i-1/2}} \right] \right\} \end{aligned}$$

$$A_{i,j} = \frac{\Delta t}{S} \frac{1}{\sigma_{i,j}^n} \frac{r_i}{(\Delta r)^2} \frac{1}{r_{i+1/2}} - u_{r_{i,j}}^{*n+1} \Delta t / 2\Delta r$$

$$D_{i,j} = \psi_{i,j}^n + \frac{\Delta t}{S} \left\{ \frac{\psi_{i,j+1}^n - 2\psi_{i,j}^n + \psi_{i,j-1}^n}{(\Delta z)^2} \right\}$$

$$- \frac{u_{z_{i,j}}^n \Delta t}{2\Delta z} \left[\psi_{i,j+1}^n - \psi_{i,j-1}^n \right]$$

Both III.26a and b can be written as

$$\tilde{A} \cdot \underline{\psi}^{n+1} = \underline{D}$$

where $\tilde{A}$ is a tri-diagonal matrix. This system is solved efficiently using the Crout's reduction method [7].

The elliptic equation in step (i), written in the finite-difference form, reduces to :

$$A_{i,j} \tilde{\psi}_{i-1,j} + B_{i,j} \tilde{\psi}_{i,j-1} + C_{i,j} \tilde{\psi}_{i,j} + D_{i,j} \tilde{\psi}_{i+1,j} + E_{i,j} \tilde{\psi}_{i,j+1} \quad (III.27)$$

where

$$A_{i,j} = \frac{-\alpha_i \beta_{i-1/2}}{\delta \rho_i \Delta \rho_{i-1}} \quad , \quad B_{i,j} = \frac{-\zeta_j^2 \zeta_{j-1/2}^2}{\delta \zeta_j \Delta \zeta_{j-1}}$$

$$D_{i,j} = \frac{-\alpha_i \beta_{i+1/2}}{\delta \rho_i \Delta \rho_i} \quad , \quad E_{i,j} = \frac{-\zeta_j^2 \zeta_{j+1/2}^2}{\delta \zeta_j \Delta \zeta_j}$$

$$C_{i,j} = -[A_{i,j} + D_{i,j}] + [B_{i,j} + E_{i,j}], \quad \alpha_i = \rho_i (1 - a \rho_i)$$

$$\beta_i = \rho_i^3 / (1 - a \rho_i) \quad , \quad \gamma_i = (1 - a \rho_i) / \rho_i$$

$$= \gamma J_{e_{i,j}}$$

$$\delta\rho_i = (\Delta\rho_i + \Delta\rho_{i-1})/2 = (\rho_{i+1} - \rho_{i-1}) / 2$$

$$\delta\zeta_j = (\Delta\zeta_j + \Delta\zeta_{j-1})/2 = (\zeta_{j+1} - \zeta_{j-1}) / 2$$

$$\zeta_{j+1/2} = (\zeta_{j+1} + \zeta_j) / 2, \text{ etc.}$$

Although the derivatives were central differenced during the derivation of III.27, e.g.,

$$\frac{\partial\psi}{\partial\rho} \rightarrow \frac{\psi_{i+1} - \psi_{i-1}}{\rho_{i+1} - \rho_{i-1}}$$

which ordinarily results in second order accurate difference equations on a uniform mesh, here a question arises as to the accuracy (or the order) of Eq. III.27, since our mesh is not exactly uniform. The original mesh in (r, z) coordinates is uniform, i.e., both Δr and Δz are constant over the computational domain in the plasma region. But the transformation to (ρ, ζ) coordinates introduces non-uniformities since

$$|\Delta\rho_i| = |\rho_{i+1} - \rho_i| = \frac{1}{r_i + a} - \frac{1}{(r_i + \Delta r) + a}$$

which gives

$$|\Delta\rho_i| = \frac{\rho_i^2 \Delta r}{1 + \rho_i \Delta r} \quad (\text{III.28})$$

which obviously is not a constant. Here we claim and will prove below that $\Delta\rho_i$ varies slowly enough so that central differencing leads to second order accurate equations.

Let us central difference the derivative $\partial\psi/\partial\rho$ on a mesh whose spacing is given by III.28:

$$\left. \frac{\partial \psi}{\partial \rho} \right|_{\rho=\rho_i} = \frac{\psi_{i+1} - \psi_{i-1}}{\rho_{i+1} - \rho_{i-1}} + E_t$$

where E_t is the truncation error. Expanding ψ_{i+1} , and ψ_{i-1} in Taylor series about ρ_i and subtracting, we get

$$\psi_{i+1} - \psi_{i-1} = (\Delta\rho_i + \Delta\rho_{i-1}) \left. \frac{\partial \psi}{\partial \rho} \right|_{\rho_i} + \frac{1}{2}[\Delta\rho_i^2 - \Delta\rho_{i-1}^2] \left. \frac{\partial^2 \psi}{\partial \rho^2} \right|_{\rho_i} + O(\Delta\rho_i^3)$$

Dividing by $\rho_{i+1} - \rho_{i-1} = \Delta\rho_i + \Delta\rho_{i-1}$ we have

$$\frac{\psi_{i+1} - \psi_{i-1}}{\rho_{i+1} - \rho_{i-1}} = \left. \frac{\partial \psi}{\partial \rho} \right|_{\rho_i} + \frac{1}{2}[\Delta\rho_i - \Delta\rho_{i-1}] \left. \frac{\partial^2 \psi}{\partial \rho^2} \right|_{\rho_i} + O(\Delta\rho_i^2)$$

Thus, central differencing seems to give only first order accurate results; but a careful inspection shows that

$$\Delta\rho_i - \Delta\rho_{i-1} \sim \frac{2(\Delta\rho_i)^2}{\rho_i}$$

Since the non-uniformity given by III.28 occurs away from the origin,

$\rho_i > 0$, and $2/\rho_i$ is bounded; therefore,

$$\Delta\rho_i - \Delta\rho_{i-1} = O(\Delta\rho_i^2)$$

Thus,

$$\frac{\psi_{i+1} - \psi_{i-1}}{\rho_{i+1} - \rho_{i-1}} = \left. \frac{\partial \psi}{\partial \rho} \right|_{\rho_i} + O(\Delta\rho_i^2)$$

i.e., as claimed above, with this "mildly" non-uniform grid-spacing, central differencing leads to second order accuracy, and the finite-difference form of the vacuum equation, Eq. III.27, is indeed second order accurate.

The solution of the difference equation III.27, ordinarily a straightforward procedure, poses some problems here because of the

nature of the coefficients involved. We will discuss these problems, and the numerical methods used in the next section.

III.4c Numerical Solution of the Vacuum Equation.

III.27 is a five-point difference equation for the elliptic vacuum equation III.21. It is to be solved on a domain with irregular boundaries, namely the computational domain of Fig. III.7b, from which the plasma region is carved out. The coefficients A,B,C,D, and E are neither symmetric, nor constant. In fact, they vary by many orders of magnitude over the domain of interest. In a way, by transforming from (r,z) to (ρ, ζ) coordinates, we have "cramped" the information about an infinite domain into a finite domain; correspondingly, the slowly varying coefficients of the equation in (r,z) coordinates are replaced now by the fast and widely varying coefficients of III.27, which have a much higher "information density" packed into them.

Because of the properties of the coefficients cited above, our attempts at solving III.27 using the successive overrelaxation method (SOR) [7] proved costly in terms of computer time. Lack of a theoretical guide for the value of the relaxation parameter for non-symmetric systems with variable coefficients forced us to adopt a trial and error procedure to find an optimum value for it. But unfortunately, this meant that we had to use the same parameter during the whole course of the iterations, whereas the optimum value actually changes as the iterations converge. This inability to follow an optimum path, coupled with the inherently slow convergence of SOR when the

coefficients vary by large amounts, forced us to look for other methods of solution.

The problem we are concerned with here is the inversion of the linear system

$$\underline{A}x = \underline{b} \quad (\text{III.29})$$

where $\underline{A}$ is a penta-diagonal band matrix of the form

$$\underline{A} = \begin{pmatrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{pmatrix}$$

The coefficients A , B , C , D , and E form the non-zero diagonals of the matrix $\underline{A}$. An efficient algorithm for the solution of systems of the type given in III.29, based on an incomplete Cholesky decomposition of $\underline{A}$, followed by conjugate gradient iterations, is described by Kershaw [14]. A subroutine (ICLU [15]) using this method to solve systems with nine diagonals is available at the MFE Computer Center in Livermore. We adapted this routine to our penta-diagonal problem in III.29 and used it in our code to solve the vacuum equation, since it proved to be much faster than SOR. For the sake of completeness we review in Appendix II the Cholesky decomposition of $\underline{A}$, and conjugate-gradient iterative solution of linear systems.

IV. RESULTS

In our simulations we were not able to reach the desired Tormac equilibrium configuration (Fig. IV.1), using the initial conditions and the boundary values described earlier. The reasons will be discussed in subsequent sections, but here we will give an overall qualitative description of the gross plasma dynamics at early times.

Typically there are three distinct phases of the initial dynamics. In the first phase, the plasma implodes both in r and z directions, driven by the rising fields at the boundary. The radial implosion is typically over in $2.4 \mu\text{sec.}$, marking the end of the first phase. Fig. IV.2 shows the density contours at $2.4 \mu\text{sec.}$ A 3-D plot of the density at the same time is shown in Fig. IV.3. The main feature of the density profile at this time is that there is a main body of the plasma near the center of the device to which two horn-like structures are attached at the top. The temperature of the pinch is a modest 10 eV, and the toroidal field profile (Fig. IV.4) shows a definite well at the center where the main body of the pinch is located.

Note that the appearance of these horns was quite unexpected, and they should not be confused with the cusp formation. As it will be explained later, the horns form basically because of an uneven rate of implosion during the early times, and the regions in the poloidal plane where they form are quite far away from the expected location of the cusps.

FIG. IV.1. External currents, and the plasma equilibrium position in Tormac.

M -1201
TIME- 2.40E-06 SEC.
DENSITY CONTOURS

FIG. IV.2. Density contours at the end of the first phase of the initial dynamics ($t = 2.4 \mu\text{sec.}$), showing the two horns and the main body of the pinch to which they are attached.

FIG. IV. 3

M -1201 TIME- $2.40E-06$ SEC.DENSITY (MAXIMUM- $4.52E+00$)

M -1201

TIME- 2.40E-06 SEC.

Z:BZ FIELD T:TOROIDAL FIELD (KG.) AT Z- 0. CM.

FIG. IV.4. Field profiles at the mid-plane at the end of the first phase.

Note the existence of a broad well in the toroidal field profile.

The second phase is characterized by the radial expansion of the overcompressed main body, and the fast inward movement of the two horns (towards the mid-plane $z=0$). The velocities at which they move scale like the Alfvén speed based on the poloidal fields, which is on the order of $5 \cdot 10^6$ cm/sec. at this time. Fig. IV.5 shows the density contours at 3.0 μ sec. The horn labelled (II) in Fig. IV.2 has merged with the main body of the pinch, whereas the one labelled (I) is still moving towards the mid-plane. At this time, mainly due to the compression from above, the temperature has risen to 14 eV in the main body. The toroidal field well is still there. By 4.0 μ sec., the horn I has moved down to the mid-plane, as shown in Fig. IV.6, and the expansion of the overcompressed main body is well underway.

In the last phase, there are two simultaneous processes leading to loss of confinement. The first is the partial disappearance of the toroidal field well, easing the expansion of the plasma towards the inner wall. As it will be explained in more detail later, we believe that the dynamics of the horns, rather than the diffusion of the fields into the plasma, leads to this partial disappearance of the well. The end result is that the plasma is confined mainly by the poloidal fields. The second process is the expansion of the plasma along the poloidal field lines. This is noticeable mainly in the cusp region, and along the field lines which run parallel to the conducting inner wall (Fig. IV.7). Fig. IV.8 shows the density and temperature profiles at 6 μ sec. at the mid-plane. The temperature is 14 eV. The plasma is practically in contact with the wall on the inside; it is kept away from the outer wall by the strong poloidal fields on the outside (Fig. IV.9).

M -1501
TIME- 3.00E-06 SEC.
DENSITY CONTOURS

FIG. IV.5. Density contours after horn II has merged with the main body of plasma.

M -4001 TIME- 4.00E-06 SEC.
 DENSITY (MAXIMUM- 5.06E+00)

FIG. IV.6. Density distribution in the poloidal plane after horn I arrives at the mid-plane.

M -6000 . TIME- 6.00E-06 SEC.
 DENSITY (MAXIMUM- 5.67E+00)

FIG. IV.7. At 6.0 μ sec. most of the plasma is at the mid-plane and near the inner wall. Notice also the plasma piling up near the outer wall at the cusp region because of the flow along the open field lines

M -6000

TIME- 6.00E-06 SEC.

D:DENSITY +:TEMPERATURE (EY.) AT Z- 0. CM.

FIG. IV.8. Density (normalized) and temperature (eV) profiles at 6.0 μ sec.

FIG. IV.9 Field profiles at 6.0 μ sec.

IV.1 Analysis of the Results.

During the design of the Tormac experiment, the external currents were chosen using a sharp-boundary model such that at their peak values, the boundary of the plasma in Fig.IV.1 would be a constant- $|B|$ surface. But this fact does not necessarily ensure that the toroidal and poloidal fields produced by these currents during the initial phase will have the correct values for the proper implosion of the plasma from its initial distribution to the configuration of Fig.IV.1. In other words, unlike the more conventional devices such as Tokamaks or various pinches, in Tormac or any other device employing an absolute minimum- $|B|$ geometry for confinement, we can not necessarily expect the fields designed for a particular equilibrium profile to push the plasma into that equilibrium configuration, if it is not already near that equilibrium state initially. In Fig. IV.10a, lowering the potential at the point (A) does not necessarily mean that all the balls will end up at the point (B) in Fig. IV.10b. This was one of the reasons why the early experiments with min- $|B|$ geometries [1],[2],[3] used plasma guns or other injection devices to force the plasma into an existing min- $|B|$ well.

The two horn-like structures mentioned earlier play a major role in the failure to implode to the desired equilibrium configuration of Fig. IV.1, starting with the uniform or semi-uniform initial conditions described in section II.4; therefore, we will examine the formation of these horns, their evolution in time, and how they adversely affect the

FIG. IV.10. Lowering the point A in (a) does not necessarily mean that all the balls will end up at the point B in (b). Some of them might get trapped in other "potential wells" during the process, as shown in (b).

initial plasma dynamics.

The horns form basically because of an uneven distribution of the field pressure around the plasma perimeter during the implosion. Fig. IV.11a shows the distribution of $|B|^2$ one grid point inside the boundary at 1μ sec. The field pressure is higher in the vicinity of the external currents which produce the poloidal fields, and naturally it is lower away from these currents, e.g., near the upper corners in Fig. IV.11a. Especially near the upper right hand corner, where the poloidal fields due to the two positive currents (I_1 and I_2) have a null point, the field pressure has quite a distinct local minimum, and as one goes away from this corner along the boundary, it shows a sharp increase. As expected, the implosion of the plasma is faster in the high field regions. Besides this unevenness of the rate of implosion, the direction of the implosion contributes to the formation of the horns. A plot of the constant- $|B|$ contours near the boundary in Fig. IV.11b shows that the field gradients are directed in such a way as to push the plasma into these horns in the direction shown by the arrows in the same figure. Thus, these two related factors, a) the uneven rate of implosion, and b) the implosion in the "wrong" direction, lead to the formation of these structures that we see in Fig. IV.2.

Of course, the horns are not stationary in time. While the majority of the initial plasma implodes to form a pinch near the center of the device, the horns move under the influence of the external fields. Now we will analyze the negative effects of their motion on the confinement. Fig. IV.12 shows the distribution of the poloidal field

FIG. IV.11a. Field pressure (B^2) one grid point inside the boundary at $1 \mu\text{sec}$.

FIG. IV.11 b. Constant $-|B|$ contours at $1 \mu\text{sec}$., showing the implosion in the "wrong" direction, which contributes to the formation of the horns.

lines around the horns at $2.4 \mu\text{sec.}$, by which time the radial implosion of the main body of the pinch is over. The field lines bend around horn II. Recalling that the magnetic field pressure tensor has a term corresponding to tension along the field lines, we expect that these bent lines will try to straighten by compressing the horn in the direction of the solid arrow in Fig. IV.12. In fact, this is what we observe in the simulations. It gets compressed in the direction shown and merges with the main body of the pinch by $3 \mu\text{sec.}$ (Fig. IV.5). Two important points to consider about this merger are:

i) It injects plasma into a region that was overcompressed to begin with by the initial radial implosion, thus accelerating the expansion of the plasma there against the toroidal field.

ii) More important is the oblique direction of the compression (the direction of the solid arrow in Fig. IV.12). The momentum imparted to the plasma in this horn during compression carries it towards a region at the mid-plane between the inner wall and the main body of the pinch; therefore, even if the main body did not expand at all due to overcompression, this process would serve as an effective expansion and bring the plasma closer to the inner wall.

Now looking at the field distribution around horn I in Fig. IV.12, we see that it also will move towards the mid-plane. The region labelled (C) in Fig. IV.12 is almost completely devoid of poloidal fields. Whatever poloidal fields that were initially present due to the initial toroidal current have been convected towards the center with the

FIG. IV.12. Poloidal field distribution around the horns at $2.4 \mu \text{sec}$. The "bent" field lines around horn II push it in the direction of the solid arrow.

FIG. IV. 13. The field pressure at point (1) is much higher than at (2) since $B_\phi(1) \approx B_\phi(2)$, but $B_p(2) \approx 0$, whereas $B_p(1) > 0$; therefore, horn I is pushed towards the mid-plane.

radial implosion. Thus, if we look at the field distribution along a vertical line cutting through this horn (Fig. IV.13), we see that although the toroidal fields are comparable in magnitude on the top and the bottom surfaces, due to the poloidal fields the field pressure is higher on the top surface, and it gets pushed downward towards the mid-plane. Due to symmetry about the mid-plane, there is also a similar structure below the r -axis moving up. The vertical implosion of these symmetric horns towards each other leads to a substantial increase in density in the region (C) in Fig. IV.13.

The dynamics of the horns that we have been describing above are not affected by the presence (or the absence) of a toroidal field well in which the main body of the pinch is located. Their motion is governed by forces away from the main pinch. However, the way the horns move does affect the toroidal field well. To see how this comes about, we will look at a 1-D case of a diamagnetic plasma which is held in equilibrium by a vertical field on the outside (Fig. IV. 14a). The forces due to the plasma pressure gradients on the inside are balanced by the forces due to the sharp gradient in the toroidal field there. But the field pressure gradients are radial, and they would not oppose the injection of plasma in the $\pm z$ directions at the location labelled (S) in Fig. IV.14a. In such a case, the toroidal field convected by the plasma that is injected there would alter the toroidal field profile and the pressure balance. In fact, this is the case in the later phases of the horn dynamics in Tormac. The plasma brought down to the mid-plane by the horns alters the toroidal field profile, as in Fig. IV.14b. The toroidal field gradient is not as steep as before (closer

FIG. IV.14. When some plasma is injected into and to the left of the region labelled S in (a), the toroidal field profile is altered because of the field convected by the plasma. The wall on the left hand side of the well is not as steep as before, which allows the plasma to expand in that direction, as in (b).

to $-1/R^2$ than before). In other words, the wall on one side of the potential well in which the plasma is confined is lowered (See Fig. IV.9), allowing the expansion of the plasma towards the inner wall, driven by the pressure gradients. Note that this process which modifies the toroidal field well occurs on a time scale determined by the rate at which the plasma moves down to the mid-plane, which is on the Alfvén-transit time scale; therefore, it is quite faster than the rate at which the well fills due to resistive diffusion.

The compression of horn I is over by about $4.0 \mu\text{sec}$. Fig. IV.6 shows the density distribution at this time. The well-defined pinch at the center has given way to a broad density distribution at the mid-plane, and the expansion along the field lines near the inner wall and the cusp region has already started. For $t > 4 \mu\text{sec}$., the main feature of the dynamics is the expansion of the plasma along the poloidal field lines, which eventually leads to the plasma distribution in the poloidal plane of Fig. IV.7.

Aside from the motion of the horns, another important reason for the failure to implode in the desired direction is a certain "lack of cooperation" between the toroidal and the poloidal fields in the confinement of the plasma. In the desired equilibrium profile of Fig. IV.1 these fields help each other. On the outside surface (large R) the poloidal field, and on the inside the toroidal field dominate, but at all points around the plasma boundary they cooperate in keeping the plasma in the equilibrium position. However, this is often not so during the implosion of the plasma in the simulations. The cusp field

lines due to the currents I_1 and I_2 (Fig. IV.15) have a negative slope in most of the plasma region in the poloidal plane because of the way the currents are distributed, namely I_2 is at lower z than I_1 . (Compare this with the positive slope of the same type of field lines in the equilibrium configuration of Fig. IV.1.) The result is that the poloidal field gradient,

$$\nabla B_p^2 \sim (+\vec{r}) \left| \frac{\partial}{\partial r} B_p^2 \right| + \vec{z} \left| \frac{\partial}{\partial z} B_p^2 \right|$$

partially opposes

$$\nabla B_\phi^2 \sim -\vec{r} \left| \frac{\partial}{\partial r} B_\phi^2 \right| + \vec{z} \left| \frac{\partial}{\partial z} B_\phi^2 \right|$$

Thus, while the toroidal field tries to push the plasma towards the equilibrium position of Fig. IV.1, the poloidal fields produced by the currents I_1 and I_2 tend to push it towards the inner wall away from that equilibrium (See Fig. IV.15).

Even at later phases of the implosion when the plasma has imploded to a narrow region around the mid-plane, in most places around the plasma the poloidal fields have an "unfavorable gradient" (not curvature), in the sense that the field pressure forces the plasma away from the desired equilibrium configuration. In Fig. IV.16, which shows some of the poloidal field lines at 4 $\mu\text{sec.}$, the plasma is compressed from the top and the bottom (Recall that we are showing only the upper half plane.) and has no where to go but to the inner wall or to the cusp region along the open field lines.

M -3201
TIME- 3.20E-06 SEC.
POLOIDAL FLUX CONTOURS

FIG. IV.15. Poloidal flux contours showing the "negative slope" of the field lines due to the currents I_1 and I_2 in most of the poloidal plane.

M -4001
 TIME- 4.00E-06 SEC.
 POLOIDAL FLUX CONTOURS

FIG. IV. 16. Some of the flux contours at 4.0μ sec, showing the compression by the poloidal fields from the top, which squeezes out the plasma towards the cusp region and the inner wall.

IV.2 Numerical Experiments

In simulations certain numerical experiments can be conducted to explore the physical system being simulated under different parameter regimes, within the framework of the physical model that is being used. In an initial-boundary value problem like ours, experimentation mainly consists of changing various initial conditions or boundary values and observing the response of the system to these various "stimuli". Of course one can also do experiments with various difference methods used to solve the equations for the sake of numerical analysis, test of accuracy, etc., but we will not discuss such experiments here.

Our main objective in conducting numerical experiments has been to find a better operating regime for Tormac, since it became obvious to us after an initial "shake-down" of the code that the plasma in Tormac was behaving in quite a different manner than the previous speculations or theoretical analyses have predicted. Believing that that the fault lied with the "start-up" phase during which the fields rise to their maximum values and the plasma equilibrium is supposed to be established, we set out to do a systematic study of various parameters to improve upon the observed dynamics in the simulations.

IV.2a Effects of Various Initial Conditions

The usual initial conditions with which the simulations start were described in section II.4. Here we will discuss the effects of various changes in those conditions on the subsequent evolution of the plasma.

i) Density and Temperature Profiles:

To prevent the initial toroidal current from self-pinching, the initial density and/or temperature profiles are made non-uniform. By making the density or temperature higher in the portion of the poloidal plane where the initial toroidal current flows, the self-pinching effects of the current can be counteracted by kinetic pressure gradients. But even in those cases where we start with completely uniform density and temperature profiles, there are no significant changes in the results for $t > 2-3 \mu\text{sec}$. During these first few microseconds, the plasma can be divided into two regions as shown in Fig. IV.17. In the shrinking region labelled I.C., the effects of the initial conditions dominate, whereas, in the region labelled B.V, the dynamics is influenced more by the boundary values at the plasma-vacuum interface. The closed curve separating the regions B.V. and I.C. shrinks at a rate determined by the characteristic signal propagation speed in the hyperbolic fluid equations, which is the Alfvén speed in our case.

At this point, we have to note that we are comparing the effects of initial conditions not too different from each other. Of course, a drastic change like increasing the initial temperature up to 1 keV would certainly make a difference in the initial dynamics, but then that would not be a reasonable initial condition with which to start our Tormac simulations!

FIG. IV. 17. At early times the plasma region can be divided into two areas. While the shrinking region labelled IC is still under the influence of the initial conditions, the region labelled BV is influenced more by the boundary values.

ii) Direction of the Initial Toroidal Current

The direction of the initial toroidal current, i.e., whether it is parallel or anti-parallel to I_3 in Fig. IV.1, makes a difference in how long the closed poloidal field lines survive during the implosion. But in either case, the rate of reconnection is fast enough so that the sense of the current does not make any significant changes, again for times longer than a few microseconds. This point will be discussed more fully in section IV.3.

iii) The Magnitude of the Toroidal Bias Field:

The magnitude of the toroidal bias field is one of the factors which affect the rate of implosion. A large B_{ϕ_0} makes the plasma stiff, and it becomes harder to implode. Since the goal is to implode the plasma and not to impede the implosion, we use the smallest value consistent with the initial conditions in the experiment, i.e., $B_{\phi_0} = 300$ Gauss. However, one beneficial effect of a large bias field is to slow down the rate of reconnection of the closed poloidal field lines, as it will be discussed in section IV.3.

IV.2b Higher External Fields

A series of runs was made with increased external currents and/or rise times, keeping the distribution of the currents in the poloidal plane constant. The temperature was higher in these runs as expected, but the confinement did not improve; the only difference here was that

the formation of the horns, their movement towards the mid-plane, expansion along the field lines, etc., happened on a faster time scale. The end result was the same as described in sections IV.1 and 1a.

IV.2c Changes in the External Current Distribution and Timing

Increasing the external currents, or having them rise faster does not alleviate the problems of Tormac; it just shortens the time scale on which they develop and lead to loss of confinement. As discussed in sections IV.1 and 1a, the biggest problems are the formation of the horns because of the non-uniformities in the field pressure around the plasma boundary at early times, and the "lack of cooperation" between the poloidal and the toroidal fields. Since the horns play a major role in the failure to implode towards the desired equilibrium configuration, we set out to prevent their formation. Below we will describe a series of changes in the external current distribution and timing which greatly improve the confinement characteristics of the plasma.

i) One source of non-uniformity in the field pressure is the poloidal field null between the two parallel currents I_1 and I_2 (Fig. IV.1), which leads to a minimum in the total field strength near the corner between I_1 and I_2 (Fig. IV.11a), which in turn is the major reason for the formation of horn II (Fig. IV.2). Letting $I_2 = 0$ gets rid of that null and moves the cusp up towards the corner. This simple modification eliminates horn II and its undesirable after-effects. It also reduces the regions where the poloidal field lines coming in from the top have negative slopes and thus causes the poloidal fields to

become more cooperative with the toroidal field. Figs. IV.18 and 19 show the density contours and field profiles, respectively, at $8 \mu \text{sec.}$, with $I_2 = 0$. Comparing with Fig. IV.7, we see that the confinement has improved a great deal; the density distribution does resemble, to some degree, the desired cusp equilibrium configuration, and the toroidal field shows a distinct well in which most of the plasma is located. The only problem here is that the plasma is too close to the inner wall, which leads to our next modification.

ii) Above the plasma is pushed towards the inner wall by the high poloidal fields on the outside (large R). In other words, the force due to the vertical field produced by I_3 more than compensates for the natural tendency of the plasma column to expand in major radius, and the radial force produced by the toroidal field well. Although theoretically it is possible to calculate the amount of poloidal field necessary to hold the plasma column in radial equilibrium, because of the complicated geometry of the Tormac equilibrium, it is easier to follow a cut and try procedure. We made a series of runs with decreasing values of I_3 until the pinch which forms after the initial implosion is in radial equilibrium, and we finally settled with a value a third of the one used in the original design. Figs. IV.20 and 21 show the density contours at two different times during the implosion with this new value of I_3 . Note that horn I has reemerged! Above where the column was right next to the inner wall, horn I was not visible because it had merged with the main pinch. Now that we decreased the poloidal field on the outside and thus kept the column off the inner wall, horn I showed itself to remind us that, although we took

M -4000
TIME- 8.00E-06 SEC.
DENSITY CONTOURS

FIG. IV.18. Density contours after we let $I_2 = 0$.

M -4000

TIME- 8.00E-06 SEC.

Z:BZ FIELD T:TOROIDAL FIELD (KG.) AT Z- 0. CM.

FIG. IV. 19. Toroidal (T), and the vertical (Z) fields at the mid-plane with $I_2 = 0$.

M -1201
TIME- 2.40E-06 SEC.
DENSITY CONTOURS

FIG. IV.20. Density contours with $I_2=0$ and reduced I_3 . Horn I has reemerged.

M -3001
TIME- 6.00E-06 SEC.
DENSITY CONTOURS

FIG. IV. 21. Horn I in Fig. IV.20 has moved down to the mid-plane leading to a considerable density increase near the inner wall.

some appropriate steps to get rid of horn II, we have not yet done anything to prevent the formation of horn I! As in the original design, it forms due to a strong non-uniformity in the field pressure near I_1 and is pushed by the poloidal fields towards the mid-plane, eventually leading to a considerable density increase between the inner wall and the main pinch as before. Next we will consider a way of dealing with horn I.

iii) The plasma feels the strong non-uniformities in the field pressure only if it is in the vicinity of the external currents. If somehow we were to push it away from these currents towards the center of the device, then the individual contributions from these currents to the field pressure would diminish, and the plasma would come under the collective influence of all the external currents, with a field pressure more uniform and closer to the equilibrium field distribution. One way of giving such a push to the plasma is to let the toroidal field do the imploding initially. The toroidal field pressure is completely uniform along the vertical walls, and it is "uniform" along the horizontal wall in the sense that it uniformly falls off in R and does not exhibit any local minima or maxima, as the poloidal fields due to the external currents do; therefore, the implosion by the toroidal field will be uniform and will not develop the horns that we see when the implosion is mainly by the poloidal fields. Thus, the method we use to prevent the formation of horn I is to keep $I_1 = 0$ for some time during which the toroidal field pushes the plasma away from the high field region near I_1 . Of course, we still need to have $I_3 \neq 0$ so that we have enough vertical field on the outside to keep the plasma from being pushed to

the outer wall by the toroidal field during this transition period.

Fig. IV.22 shows the new timing for the external currents. I_{ϕ_0} , which produces the toroidal field, and I_3 come on at the same time, with the old rise time of $8 \mu\text{sec.}$ But I_1 is delayed $2.5 \mu\text{sec.}$, after which it rises in $5.5 \mu\text{sec.}$ to its maximum value. We let I_1 rise relatively faster than the others to have all the external currents at their maximum values at $8.0 \mu\text{sec.}$ so that we can watch the evolution of the plasma with fixed external currents.

Fig. IV.23 shows the new external current arrangement in the poloidal plane. As discussed earlier, I_2 is no longer there. We also placed I_1 closer to the inner wall than in the original design to help the toroidal field push the plasma away from the inner wall.

Our success at eliminating horn I becomes evident when we look at density contours at $4.8 \mu\text{sec.}$ in Fig. IV.24. The plasma profile in the poloidal plane resembles that of the desired cusp equilibrium (Fig. IV.1), with no trace of horn I. There is a well formed pinch at the center, with plasma extending towards the cusp along the open field lines. Looking at the field profiles (Fig. IV.25), we see that the plasma is located in a quite visible toroidal field well. More importantly, comparing constant- $|B|$ contours (Fig. IV.26) with the density contours in Fig. IV.24, we see that the plasma is located in a minimum $-|B|$ well. Thus, at this instant in time the plasma seems to be in a state which has all the desired characteristics of a cusp equilibrium, except that we do not have equilibrium yet! The

FIG. IV.22. New external current profiles.

FIG. IV.23. New external current distribution.

M -4801
TIME- 4.80E-06 SEC.
DENSITY CONTOURS

FIG. IV.24. Density contours at 4.8μ sec. with the new external current distribution and timing shown in Figs. IV.22 and 23.

M -4801

TIME- 4.80E-06 SEC.

Z:BZ FIELD T:TOROIDAL FIELD (KG.) AT Z- 0. CM.

FIG. IV. 25. Field profiles at 4.8 μ sec. showing the toroidal field well in which most of the plasma in Fig. IV.24 is located.

M -4801
TIME - 4.80E-06SEC.
CONSTANT ABS(B) CONTOURS

FIG. IV.26. Constant $|B|$ contours showing the min- $|B|$ region at the center where the major portion of the plasma is located at this time.

macroscopic velocities are still quite large ($5 \cdot 10^6$ cm/sec), and the plasma profiles are still evolving. The density contours at 8 μ sec. have two important features (Fig. IV.27): First of all, notice that the plasma is leaving the body of the pinch (the central region), and it is flowing along the open field lines towards the cusp region and piling up against the outer wall near the upper right hand corner. Secondly, we see a new "ear" forming near the inner wall, again due to plasma flow along the open field lines. We will discuss such motions along the field lines in the next section.

IV.2d Motion Along the Open Field Lines

We can explain the new "ear" formation and the fast flow towards the cusp region in terms of the plasma motion along the open poloidal field lines. As is well known from Tokamak equilibrium theory, in static equilibrium, the plasma can not support any pressure gradients along the poloidal field lines, since

$$(\underline{B} \cdot \nabla)p = \underline{B} \cdot \frac{1}{c} (\underline{J} \times \underline{B}) = 0 \quad (\text{IV.1})$$

which, because of axisymmetry, gives

$$\left(\frac{B_p}{r} \cdot \nabla \right) p = 0 \quad (\text{IV.2})$$

Thus, in our scalar pressure MHD model, the ear formation and the fast flow towards the cusps can be interpreted as the attempts by the plasma to equalize the pressure along the open poloidal field lines of Fig. IV.28.

M -8001
TIME- 8.00E-06 SEC.
DENSITY CONTOURS

FIG. IV.27. Density contours in Fig. IV.24 evolve to this configuration by $8 \mu\text{sec}$. Note the formation of an "ear" near the inner wall, and the plasma piling up against the outer wall near the cusp.

M -8001
TIME- 8.00E-06 SEC.
POLOIDAL FLUX CONTOURS

FIG. IV.28. Some of the poloidal field lines showing the convergence of the lines near the inner wall and towards the cusp. Notice the closed lines at the center which have survived until this time.

One of the drawbacks of our isotropic pressure assumption is that the mirror effect along converging field lines is neglected. Physically, one would expect the flow along the open field lines to be impeded by this mirror effect, both near the inner wall and the cusp region, since the field lines converge in both locations (Fig. IV.28). To be able to take into account this effect, we have to abandon our isotropic pressure assumption and adopt, for example, the Double Adiabatic Theory of Chew, Goldberger, and Low [4]. In the CGL theory, the static equilibrium condition for an axisymmetric device reduces to

$$(\underline{B}_p \cdot \nabla) p_{\parallel} = (p_{\parallel} - p_{\perp}) \frac{1}{|\underline{B}|} (\underline{B}_p \cdot \nabla) |\underline{B}| \quad (\text{IV.3})$$

The right hand side of Eq. IV.3, which drops out for the isotropic case ($p_{\parallel} = p_{\perp}$), properly takes care of the mirror effect and makes it possible for the plasma to support a pressure gradient along the open field lines. Thus, had we used this more sophisticated model, we may not have seen the ear formation, and the rate of flow to the cusp region would certainly be slower.

Why did we not use this more accurate model? If our emphasis were on the study of the cusp equilibrium state, and the related problems of the rate of flow out through the cusps, etc., then we would have to go to the more accurate CGL theory. But here we are mainly concerned with the initial dynamics during the implosion of the plasma towards that equilibrium state, and a simple scalar pressure model is quite adequate. Moreover, there is also the question of whether any anisotropy can be sustained at these low temperatures (10-15 eV), when the plasma is quite collisional. A typical temperature isotropization time [5] for ions is

given by

$$\tau^i = 5.0 * 10^7 T^{3/2} / n \ln \Lambda \quad (\text{IV.4})$$

For $n = 5 * 10^{15}$, $T = 10$ eV, $\ln \Lambda = 10$,

$$\tau^i = 3 * 10^{-8} \text{ sec.}$$

which is far shorter than the characteristic Alfvén transit time in the experiment. On the basis of such a small τ^i , we probably can not expect any anisotropy in the kinetic pressure, and our scalar pressure assumption should be quite valid. Of course, this implies that at these low temperatures, we can not expect any mirror confinement in the cusp region, or near the inner wall, and the formation of the ear and the fast flow-out to the cusp region that we observe in the simulations may have a more physical basis than one might have initially expected.

IV.3 Disappearance of the Closed Poloidal Field Lines

The closed poloidal field lines produced by the initial toroidal current open up during the course of the implosion at a rate much faster than expected from resistive diffusion. This is especially apparent when the initial toroidal current is in the same direction as the external current I_3 (See Fig. IV.1). In this case the closed field lines completely disappear in about 3 $\mu\text{sec.}$, which is far shorter than the resistive diffusion time, even at the low temperatures (3-5 eV) observed in the simulations during the first few microseconds of the implosion. As the current I_3 increases, the closed field lines break open at an X-point at the mid-plane between the plasma current and I_3 .

and reconnect with the field lines produced by I_3 , as shown in Fig. IV.29.

Some important characteristics of this field line reconnection are:

i) The rate of reconnection is not a strong function of electrical conductivity; increasing the conductivity by an order of magnitude decreases the reconnection rate, but by a much smaller factor.

ii) The rate also depends on the rate of implosion; slowing down the implosion by putting in a large toroidal bias field also slows down the reconnection rate.

iii) Finally, artificially damping the velocities by putting large numerical dissipation into the momentum equations also slows down the rate of reconnection.

All these factors suggest that, besides diffusion, inertial forces play an important role in the reconnection process.

Outside of the fusion research community, the magnetic field line reconnection, or "flux annihilation", has been investigated, especially in relation to solar flares, since late 1950's. Vasyliunas [6] gives a unifying review of various theories put forward to explain the solar flares in terms of rapid conversion, at an X-point, of magnetic field energy into the kinetic energy of the escaping gases from the solar chromosphere.

M - 401
TIME - 8.02E-07 SEC.
POLAR FLUX CONTOURS

FIG. IV.29. Reconnection of the initially closed field lines.

In the magnetic fusion community, such topology changing processes have been investigated under the name of "tearing modes". Since the classic paper by Furth, Killeen, and Rosenbluth [7] on resistive instabilities, this topic has been studied extensively. In particular, the $m = 1$ resistive tearing mode (or the resistive internal kink mode), believed to be responsible for the so called "saw-tooth oscillations" [8] in Tokamaks, has received a great deal of attention in recent years.

Rather than going into the details of this research, we will focus on some points pertinent to our discussion of Tormac. Kadomtsev [9], in his heuristic argument for the non-linear development of the $m=1$ tearing mode in Tokamaks, discusses a mechanism by which the field lines break and reconnect at the tips of the $m=1$ island at a rate that could be much faster than resistive diffusion. Shafranov [10] makes use of basically the same mechanism in his proposal for a new tokamak heating method, which is based on the mixing of separate plasma columns with parallel currents. Below we will review the relevant parts of his argument, with some modifications, to explain the rapid disappearance of the closed field lines in Tormac.

IV.3a A Heuristic Argument for the Rate of Reconnection

Initially without any external currents, the poloidal field lines in Tormac have the geometry shown in Fig. II.6. The characteristic "skin time", or the time it takes for the closed field lines to disappear due to ohmic dissipation of the current I_{ϕ_0} in the absence of

external currents, is given by

$$\tau_s = \frac{4\pi\sigma a^2}{c^2}$$

where σ is the electrical conductivity, and a is a characteristic transverse dimension of the plasma column. (We take it to be the half-width of the vacuum vessel.) Even at low temperatures, τ_s can be in tens of microseconds, due to its quadratic dependence on the length scale. As we will see below, an important effect of non-zero external currents ($I_3 < 0$) is to decrease the relevant length scale from a to a much smaller value.

Shafranov's analysis is about an even number of plasma columns with parallel currents which are pushed towards each other by external forces. He investigates the rate of annihilation of the magnetic flux between the two magnetix axes, in the case of two columns only, which eventually leads to an increase in the internal energy of the plasma. In Tormac we do not have separate columns, but the combination of the toroidal current carrying initial plasma column, and the rapidly increasing external toroidal currents (e.g. I_3 , when $I_{\phi_0} \parallel I_3$) may be considered, for our purposes, as two separate columns rapidly approaching each other. Thus, we can apply Shafranov's argument. When the external current I_3 is turned on and starts rapidly increasing (rise time = 8 μ sec.), it induces image currents in the plasma (Fig. IV.30), and the topology of the field lines starts changing (Here we are assuming $I_{\phi_0} \parallel I_3$). The image currents, which are in opposite direction to both I_{ϕ_0} and I_3 , flow in a layer of thickness δ and length a (Fig. IV.31). This is the region where the actual

FIG. IV.30. Field lines and the currents at the beginning of the reconnection process when $I_{\phi_0} \parallel I_3$.

FIG. IV.31. Current layer around the X-point in which the reconnection takes place. Field lines carried into the layer with velocity v break and reconnect at the X-point and are carried away with velocity u .

"dissipation" takes place now. The field lines in opposite directions are brought in from the outside region, which for our purposes can be regarded as perfectly conducting, with the velocity v ; they break and reconnect at the X-point and are carried away from this resistive region with the velocity u .

The velocity v is a measure of the rate of reconnection; the higher v is, the faster the reconnection takes place. But v is also related to the rate of dissipation in this resistive layer of thickness δ . The time it takes for two lines on the opposite sides of the layer to diffusively approach each other and break and reconnect at the X-point is given by

$$t_{\delta} = \frac{4\pi\sigma(\delta/2)^2}{c^2} \quad (\text{IV.5})$$

which is the time for a field line at one edge to diffusively move a distance of $\delta/2$ to the X-point. Thus, the diffusion velocity, or the velocity with which the field lines are brought together is

$$v = \frac{\delta/2}{t_{\delta}} = \frac{c^2}{2\pi\sigma\delta} \quad (\text{IV.6})$$

Notice that v depends on δ , and we have not said anything about the layer thickness yet. A second relation we need to determine δ and v is the equation of mass continuity:

$$\frac{1}{\rho} \frac{d\rho}{dt} = - \nabla \cdot \underline{u} \quad (\text{IV.7})$$

During the implosion phase, the left hand side of IV.7 defines a characteristic compression rate $1/\tau_c$. The right hand side can be estimated as (Fig. IV.31)

$$\nabla \cdot \underline{u} \approx \frac{-2v}{\delta} + \frac{2u}{a}$$

Thus, we have

$$\frac{2u}{a} - \frac{2v}{\delta} = -1/\tau_c$$

or

$$va - \delta u = a\delta / 2\tau_c \quad (\text{IV.8})$$

Note that if we assumed incompressible plasma, as Shafranov does, $\tau_c \rightarrow \infty$, and we would simply have $va - u\delta = 0$, but here we are going to keep τ_c finite, since during the implosion, the incompressible plasma assumption is not valid.

In Eq. IV.8, u is the velocity with which the reconnected field lines are carried away from the resistive region. An order of magnitude estimate for u is given by the magnetosonic speed based on the poloidal fields

$$u \approx \left[\frac{B_p^2}{4\pi\rho} + \frac{\gamma p}{\rho} \right]^{1/2} \quad (\text{IV.9})$$

where B_p is a characteristic poloidal field strength in the problem.

Using Eqs. IV.5, 6, and 8, for v we get

$$v = \left[\frac{c^2 (u + a / 2 \tau_c)}{2\pi\sigma a} \right]^{1/2} \quad (\text{IV.10})$$

The total reconnection time, or the time for all the closed field lines due to the initial toroidal current to be carried to the X-point with velocity v , is

$$t_m = a / v = \left[\frac{2 \pi \sigma a^3}{c^2 (u + a / 2\tau_c)} \right]^{1/2} \quad (\text{IV.11})$$

which can be written as

$$t_m = \sqrt{\tau_s \tau_a} / \sqrt{2 + \tau_a / \tau_c}$$

where $\tau_s = 4\pi\sigma a^2/c^2$, the "skin" time, $\tau_a = a / u$, Alfven transit time based on the poloidal field, and τ_c is a characteristic compression time during the implosion.

Some typical values for Tormac during the early times we are concerned with are:

a : 8 cm.,

u : $5 \cdot 10^6$ cm/sec.,

τ_c : 2 μ sec.,

τ_s : 50 μ sec. (for $T = 3\text{eV}$, and classical conductivity)

τ_a : 1.6 μ sec., and

t_m : 5 μ sec.

Note that t_m , the total reconnection time, is an order of magnitude smaller than the resistive diffusion time τ_s , and at least has the same order of magnitude as the time it takes for all the field lines to open up ($\sim 3 \mu$ sec.) in the simulations.

As mentioned earlier, this reconnection process is so much faster than resistive diffusion because the scale length in the problem is reduced from a , the radius of the plasma column, to δ , the width of the current layer around the X-point. In other words, the physical

mechanism behind the reconnection process is still resistive diffusion, but the diffusion length in this case has been reduced from a to δ .

The best evidence for the existence of this current layer is a "bump" in the temperature profile due to ohmic dissipation of the image currents. Figs. IV.32 and 33 show the density and temperature profiles at $1.2 \mu\text{sec.}$ from two different runs. The runs were identical in every aspect, except that in one of them (Fig. IV.32), the initial plasma current was parallel to I_3 , giving rise to a rapid field line opening (Fig. IV.29), whereas in Fig. IV.33, these two currents were antiparallel. The profiles are quite similar, except for quite a distinct bump in the temperature profile near the outer wall in the case with I_{ϕ_0} parallel to I_3 .

So far we have been discussing the case in which the field line opening was the fastest ($I_{\phi_0} \parallel I_3$). Since the cusp currents have opposite directions, no matter what its direction, the initial toroidal current I_{ϕ_0} is parallel to one of the external currents, and consequently there is always an X-point in some part of the plasma region where the reconnection process takes place. If I_{ϕ_0} is not parallel to I_3 , then it is parallel to I_1 , and I_2 (Fig. IV.1). In this case, the X-point is not at the mid-plane ($z = 0$), but there are two X-points, one above, and one below the mid-plane near the currents I_2 (Fig. IV.34). Here the total time for all the closed field lines to disappear is longer ($t_m = 5 \mu\text{sec.}$) for two reasons. First of all, the reconnection process at the X-point proceeds more slowly, because $|I_2| < |I_3|$, and consequently the induced image currents in the plasma

M - 601

TIME- 1.20E-06 SEC.

O:DENSITY +:TEMPERATURE (EV.) AT Z- 0. CM.

FIG. IV.32. Density and temperature profiles at 1.2 μ sec. when the initial toroidal current I_{ϕ_0} is parallel to the external current I_3 . Notice the bump in the temperature profile near the outer wall and compare this with Fig. IV.33.

M - 601

TIME- 1.20E-06 SEC.

D:DENSITY +:TEMPERATURE (EV.) AT Z= 0. CM.

FIG. IV.33. Density and temperature profiles when I_{ϕ_0} is antiparallel to I_3 .

M - 801
 TIME - 8.01E-07 SEC.
 POLOIDAL FLUX CONTOURS

FIG. IV.34. Flux contours when the initial toroidal current I_{ϕ_0} is antiparallel to I_3 . In this case there are two X-points, one in each half-plane.

(and the electric field) are smaller. In terms of our previous discussion, this implies a wider diffusion region around the X-point (larger δ) thus slower reconnection rate. The second reason has to do with the fact that the device is elongated in the z-direction (height to width ratio = 1.5), and the external currents I_1 , and I_2 are farther away from the plasma current than I_3 is; therefore, the characteristic length scale a , and the skin time τ_s are longer in this case, leading to a longer total reconnection time.

In any case, there is a definite reconnection process which takes place independent of the direction of the initial toroidal current. By making it antiparallel to I_3 (and parallel to I_1 , and I_2) we do not prevent its demise but delay it somewhat.

Our discussion up to now has been about the reconnection process in the original Tormac design. One of the side effects of the modifications we made to the external current distribution and timing is that the closed field lines survive quite a bit longer. Some of them are still visible at 8 μ sec. in Fig. IV.28, which shows the flux contours from a run with all the previously mentioned modifications included. The main reason is that with the current distribution and timing shown in Figs. IV.22 and 24, if $I_{\phi 0}$ is antiparallel to I_3 , there is no X-point in the plasma for the first 2.5 μ sec. of the implosion, during which time I_1 is zero, and the toroidal field is doing the imploding. During these 2.5 μ sec., the closed field lines disappear only through ordinary resistive diffusion, which occurs on a slow time scale, and not through the reconnection we talked about

earlier. When I_1 is turned on at $2.5 \mu\text{sec.}$, the plasma has already imploded away from the vicinity of this current. Consequently, the image currents (and the electric field) induced on the plasma boundary are smaller than what they would be if the plasma were resting against the wall near I_1 when it is turned on. Thus, the reconnection process is slower, due to smaller electric fields, than it would be if I_1 were not delayed in time for $2.5 \mu\text{sec.}$, leading to a longer survival time for the closed poloidal field lines.

Of course, if the initial toroidal current is parallel to I_3 , the closed field lines disappear in about $3 \mu\text{sec.}$ as before, even in this modified case, since an X-point exists at the mid-plane between I_{ϕ_0} and I_3 starting from $t = 0$, as in the case with the original Tormac design.

REFERENCES

CHAPTER I

- [1] J. Berkovitz, K. O. Friedrichs, H. Goertzel, H. Grad, J. Killeen, E. Rubin, in Proceedings of the Second United Nations Conference on Peaceful Uses of Atomic Energy (United Nations, Geneva, 1958), Vol. 31, p. 171.
- [2] J. Berkovitz, H. Grad, H. Rubin, in Proceedings of the Second United Nations Conference on Peaceful Uses of Atomic Energy (United Nations, Geneva, 1958), Vol. 31, p. 177
- [3] L. A. Spalding, Nuclear Fusion 8, 161 (1968).
- [4] M. G. Haines, Nuclear Fusion 17, 811 (1977).
- [5] M. A. Levine, A. H. Boozer, G. Kalman, P. Bakshi, Phys. Rev. Lett. 28, 1323 (1972).
- [6] A. H. Boozer, M. A. Levine, Phys. Rev. Lett. 31, 1287 (1973)
- [7] M. A. Levine, A. H. Boozer, W. B. Kunkel, in Proceedings of the High Beta Workshop, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1975.
- [8] H. Grad, Phys. Rev. Lett. 4, 222 (1960).
- [9] W. Grossmann, Phys. Fluids 9, 2478 (1966).
- [10] D. L. Book, "NRL Plasma Formulary", Naval Research Laboratory, Washington, D.C. 20375, 1978.
- [11] J. H. Hammer, W. B. Kunkel, Phys. Fluids 19, 438 (1976).
- [12] M. A. Levine, I. G. Brown, C. C. Gallagher, in Pulsed High Beta Plasmas, (D. E. Evans, ed.) Pergamon Press, Oxford, 1976, p. 387.

- [13] M. A. Levine, Bull. Am. Phys. Soc. 17, 1040 (1972).
- [14] C. C. Gallagher, M. A. Levine, Air Force Cambridge Research Laboratory, Technical Report No: 74-0122 (1974).
- [15] B.R. Myers, M.A. Levine, Rev. Sci. Instrum. 49, 610 (1978).
- [16] M. Greenwald, J. Coonrod, M.A. Levine, Bul. Am. Phys. Soc. 23, No. 7, 859 (1978).

CHAPTER II

- [1] See for example, G. Bateman, "MHD Instabilities", The MIT Press, Cambridge, Massachusetts (1978), p. 67.
- [2] H. C. Lui, C. K. Chu, Phys. Fluids 19, 1947 (1976).
- [3] C. K. Chu, and H. Grad, in "Research Frontiers in Fluid Dynamics", Interscience (1964), p. 308.
- [4] G. F. Carrier, and C. E. Pearson, "Partial Differential Equations", Academic Press, New York (1976), Chapter 13.
- [5] H. C. Lui, and C. K. Chu, Phys. Fluids 18, 1277 (1975).
- [6] W. Park, Ph.D. Thesis, Columbia University (1978).

CHAPTER III

- [1] J. Killeen, in "Physics of Hot Plasmas", (R. J. Rye, and J. C. Taylor, Eds.), Oliver and Boyd, Edinburgh (1970).
- [2] K. V. Roberts, and D. E. Potter, Methods Compt. Phys. 9, 333 (1970).
- [3] J. U. Brackbill, Methods Compt. Phys. 16, 1 (1976).
- [4] N. Sharky, Ph.D. Thesis, Columbia University (1978).
- [5] W. Park, Ph. D. Thesis, Columbia University (1978).

- [6] R. D. Richtmyer, and K. W. Morton, "Difference Methods for Initial Value Problems.", Wiley (Interscience), New York (1967).
- [7] See for example, E. L. Wachspress, "Iterative Solution of Elliptic Systems.", Prentice Hall, Englewood Cliffs, N.J. (1966).
- [8] C. K. Chu, and A. Sereny, J. Computational Phys. 15, 476 (1974).
- [9] D. Gottlieb, and E. Turkel, J. Computational Phys. 26, 181 (1978).
- [10] A. Lapidus, J. Computational Phys. 2, 154 (1967).
- [11] K. Lackner, Computer Physics Communications, 12, 33 (1976).
- [12] B. McNamara, Methods Compt. Phys. 16, 211 (1976).
- [13] A. Aydemir, and C. K. Chu, Annual Controlled Fusion Theory Conference, Gatlinburg, Tennessee, paper A18 (1978).
- [14] D. S. Kershaw, J. Computational Phys. 26, 43 (1978).
- [15] T. Cutler, ICLJ, The ICCG Solver, No:42 in LIBRIS, Magnetic Fusion Energy Computer Center (MFECC), Lawrence Livermore Laboratory, University of California (1978).

CHAPTER IV

- [1] S. Lukyanov, I.M. Podgornyi, Atomnaya Energiya, 11, 336 (1961); J. Nucl. Energy, Part C, 4, 135 (1964).
- [2] T.K. Allen, A.J. Cox, I.J. Spalding in Plasma Physics and Controlled Nuclear Fusion Research (Proc. Second Int. Conf. Culham, 1965), IAEA, Vienna, 2, 427 (1966).
- [3] D.C. Hagerman, Nucl. Fusion, Supplement Part 1, 75 (1962).
- [4] G.F. Chew, M.L. Goldberger, and F.F. Low, Proc. Roy. Soc. (London), A326, 112 (1956).
- [5] D.L. Book, "NRL Plasma Formulary", Naval Research

Laboratory, Washington, D.C. (1978).

- [6] V.M. Vasyliunas, Rev. Geophys. Space Phys. 13, 303 (1975).
- [7] H.P. Furth, J. Killeen, M.N. Rosenbluth, Phys. Fluids, 6, 459 (1963).
- [8] S. Van Goeler, W. Stodiek, N. Sauthoff, Phys. Rev. Lett. 33, 1201 (1974).
- [9] B.B. Kadomtsev, Sov. J. Plasma Phys. 1, 389 (1975).
- [10] V.D. Shafranov, Nucl. Fusion, 19, 187 (1979).

APPENDIX I

Implementation of the Continuity Conditions

on the Plasma-Vacuum Boundary

In steps (ii) and (iv) of the algorithm for the solution of the equations for the poloidal flux (Sect. III.4b), values of ψ on the plasma-vacuum boundary have to be calculated using the continuity conditions

$$[\psi] = 0, \quad \left[\frac{\partial \psi}{\partial n} \right] = 0.$$

Here we will show how we can obtain a third-order accurate equation for the flux function on the boundary using these conditions.

To avoid the complications that would arise from taking left-hand derivatives in (r, z) , and right-hand derivatives in (ρ, ζ) coordinates, etc., we will work only in (ρ, ζ) coordinates. Thus, $\psi_{i,j}$ will mean $\psi(\rho_i, \zeta_j)$, and if the point (ρ_i, ζ_j) for instance is on the vertical boundary

$$\rho = \frac{1}{r_2 + a}, \quad \frac{1}{z_1 + b} \leq \zeta \leq \frac{1}{b}$$

then $\psi_{i-1,j}$ will be in the vacuum, and $\psi_{i+1,j}$ will be in the plasma regions (See Fig. III.7).

To be able to calculate a third-order accurate value for $\psi_{i,j}$, we have to find the left and right-hand normal derivatives at (ρ_i, ζ_j) to an accuracy of $O(\Delta\rho^2)$. For example, to get the right-hand normal

derivative, we expand $\psi_{i+1,j}$ and $\psi_{i+2,j}$ about (ρ_i, ζ_j) in a Taylor series:

$$\psi_{i+1} = \psi_i + \Delta\rho_i \left. \frac{\partial\psi}{\partial\rho} \right|_{\rho_i} + \frac{\Delta\rho_i^2}{2} \left. \frac{\partial^2\psi}{\partial\rho^2} \right|_{\rho_i} + O(\Delta\rho_i^3) \quad (\text{A1.1})$$

$$\psi_{i+2} = \psi_i + a\Delta\rho_i \left. \frac{\partial\psi}{\partial\rho} \right|_{\rho_i} + \frac{a^2\Delta\rho_i^2}{2} \left. \frac{\partial^2\psi}{\partial\rho^2} \right|_{\rho_i} + O(\Delta\rho_i^3) \quad (\text{A1.2})$$

where we have dropped the subscript "j", since it does not vary, and

$$a = \frac{\rho_{i+2} - \rho_i}{\rho_{i+1} - \rho_i} = \frac{\Delta\rho_i + \Delta\rho_{i+1}}{\Delta\rho_i}$$

which would be equal to two on a uniform mesh.

Subtracting A1.2 from a^2 times A1.1, we get

$$a^2 \psi_{i+1} - \psi_{i+2} = \psi_i (a^2 - 1) + a(a-1)\Delta\rho_i \left. \frac{\partial\psi}{\partial\rho} \right|_{\rho_i} + O(\Delta\rho_i^3)$$

which gives

$$\left. \frac{\partial\psi}{\partial\rho} \right|_{\rho_i} = \frac{a^2 \psi_{i+1} - \psi_{i+2} - (a^2 - 1) \psi_i}{a(a-1)\Delta\rho_i} + O(\Delta\rho_i^2) \quad (\text{A1.3})$$

Similarly, for the left-hand normal derivative we get

$$\left. \frac{\partial\psi}{\partial\rho} \right|_{\rho_i} = \frac{-b^2 \psi_{i-1} + \psi_{i-2} + (b^2 - 1) \psi_i}{b(b-1)\Delta\rho_{i-1}} + O(\Delta\rho_{i-1}^2) \quad (\text{A1.4})$$

Equating the two derivatives, and solving for ψ_i , we obtain

$$\begin{aligned} \psi_i = & - \{ a\Delta\rho_i [\psi_{i-2} - b^2\psi_{i-1}] / (b-1) + b\Delta\rho_{i-1} [\psi_{i+2} - a^2\psi_{i+1}] / (a-1) \} \\ & / \{ a(b+1)\Delta\rho_i + b(a+1)\Delta\rho_{i-1} \} + O(\Delta\rho^3) \end{aligned}$$

$$b = (\Delta\rho_{i-1} + \Delta\rho_{i-2}) / \Delta\rho_{i-1}$$

which is the equation III.24 used in step (ii) in section III.4b. The corresponding equation for the values on the horizontal boundary, Eq. III.25 in section III.4b, is derived in a similar fashion.

Note that a first-order accurate finite differencing of the one-sided derivatives leads to a simpler, but second-order accurate expression for $\psi_{i,j}$. We use the more accurate Eqs. III.24 and 25, since they give faster convergence for the boundary values.

APPENDIX II

Cholesky Decomposition and Conjugate Gradient Iterations

In this appendix we will review the Cholesky decomposition of M , where M is a positive definite symmetric matrix, and discuss the solution of the linear system $Mx=b$ using Hestenes-Stiefel conjugate-gradient method [1].

a) Incomplete Cholesky Decomposition

A symmetric, positive definite matrix can always be written in the form [2]

$$M = LL^t = L^t L$$

where L is a lower triangular matrix. Letting

$$M_{i,j} = \sum_{k=1}^n L_{ik} L_{kj}^t = \sum_{k=1}^n L_{ik} L_{jk}$$

and recalling that $L_{ik} = 0$ for $k > i$, we can show that

$$\begin{aligned} L_{kj} &= \{M_{kj} - \sum_{p=1}^{j-1} L_{kp} L_{jp}\} / L_{jj}, \quad k < j \\ L_{kk} &= \{M_{kk} - \sum_{p=1}^{k-1} L_{kp}^2\}^{1/2} \end{aligned} \quad (A2.1)$$

A simple derivation of A2.1 is given in Gastinel [3]. If M is a large sparse matrix, as it is in our case, a complete Cholesky decomposition as defined by A2.1 proves disadvantageous, since sparseness is lost during the decomposition, with most of the elements of L being non-zero. Kershaw [4] does an incomplete Cholesky decomposition, where he forces L to have the same sparsity pattern as M , i.e., during the decomposition, he lets $L_{ij} = 0$ if $M_{ij} = 0$. Thus, he gets an approximate decomposition.

$M = LL^t + E$, where E is some error matrix. He argues that E should be negligible for problems involving parabolic or elliptic partial differential equations where M^{-1} (the Green's function) couples a point closely only to its nearest neighbors.

b) Conjugate-Gradient Method

Conjugate-gradient method, proposed originally by Hestenes and Stiefel [1], is an iterative algorithm for the solution of $Mx=b$, M being a symmetric, positive definite $n \times n$ matrix. As we will show later the method is theoretically guaranteed to converge to the exact solution in a maximum number of n iterations.

Before describing the method, we will review some pertinent points from linear algebra. Let x_0 be the solution of $Mx=b$, $x \in R^n$, and define a scalar-valued function of x ,

$$F(x) = \frac{1}{2}(x, Mx) - (x, b)$$

where (x, b) is the scalar product of x and b . Note that

$$F(x) = \text{constant}$$

is a quadratic form (hyperquadric) in the components of the vector x .

$F(x)$ has the following properties [3]:

i) $\nabla F(x) = Mx - b$. Thus, for any $x, x \in R^n$, $\nabla F(x)$ is equal to the residual $r(x) = Mx - b$. In particular,

$$r(x_0) = \nabla F(x_0) = Mx_0 - b = 0.$$

ii) The solution x_0 of $Mx=b$ is the center of one of the

hyperquadrics Q_k ,

$$Q_k : F(x) = (1/2) * (x, Mx) - (x, b) = k \quad (A2.2)$$

To show this, let $x = x_0 + tu$, where t is a scalar and u is some vector in R^n . Then x is on Q_k if and only if $F(x) = k$, or substituting for x in A2.2 we get

$$(1/2) * t^2 (u, Mu) + tu(Mx_0 - b) + F(x_0) - k = 0 \quad (A2.3)$$

where we made use of the self-adjointness of M , i.e. $(u, Mx_0) = (x_0, Mu)$.

Since $Mx_0 - b = 0$, and assuming $(u, Mu) \neq 0$, we have

$$t = \pm [2(k - F(x_0)) / (u, Mu)]^{1/2} \quad (A2.4)$$

Thus, for any $u \neq 0$, there are two points on the quadric Q_k , symmetric to $x = x_0$,

$$x_{1,2} = x_0 \pm [2(k - F(x_0)) / (u, Mu)]^{1/2} u$$

In other words, x_0 is the (unique) center of symmetry of Q_k .

iii) Q_k defined in A2.2 is a hyper-ellipsoid. This follows from the positive-definiteness of M , $(u, Mu) > 0$ for $u \neq 0$; thus,

$$\begin{aligned} F(x_0) &= (1/2) (x_0, Mx_0) - (x_0, b) = (1/2) (x_0, Mx_0) - (x_0, Mx_0) \\ &= -(1/2) (x_0, Mx_0) < 0. \end{aligned}$$

and for $k > 0$, t as given by A2.4 has two distinct real solutions, which implies Q_k is an ellipsoid. In summary, x_0 , the solution of $Mx=b$, where M is a symmetric, positive definite matrix, is the center of one of the system of hyper-ellipsoids defined by Eq. A2.2.

iv) The solution of $Mx=b$ is also the point x_0 in R at which

$$F(x) = (1/2)*(x, Mx) - (x, b)$$

attains its minimum. This follows from Eq. A2.3, which can be written in the form

$$F(x) = F(x_0) + (1/2)*t^2*(u, Mu),$$

where $x = x_0 + tu$ again, and $Mx_0 = b$. Since M is positive definite, $F(x)$ is minimum when $t = 0$, the minimum value being given by $F(x) = F(x_0)$.

The conjugate-gradient method involves taking various "cuts" through this ellipsoid, much like chopping pieces off an apple, such that one eventually takes a cut through the dead-center of the ellipsoid, which yields the exact solution.

The algorithm, which we take from Gastinel [3], proceeds as follows:

Let $x_0 \in R^n$ be an initial guess for the solution of $Mx = b$. Then the residual r_0 ,

$$r_0 = Mx_0 - b = \nabla F(x_0)$$

points in the direction of fastest increase in F . As in the method of steepest descent, we take x_1 to be in the direction of $-r_0$ from x_0 . More precisely, x_1 is chosen such that $-r_0$ is tangent to the quadric through x_1 ,

$$F(x) = (1/2)*(x, Mx) - (x, b) = F(x_1), \quad x \in R^n,$$

at the point x_1 .

Then we find $r_1 = \nabla F(x_1)$. Now the vectors r_0 and r_1 define a plane P , and the intersection of this plane with the ellipsoid defined by $F(x) = F(x_1)$ is a closed conic section C of Fig. A2.1. The equation

of this conic section is

$$F(x) = (1/2) * (x, Mx) - (x, b) = F(x_1), \quad x \in P \quad (A2.5)$$

Now let $u_0 = -r_0$. The vector u_0 determined a search direction from x_0 in our search for the point x_1 . We take our new search direction u_1 to be the direction of the center of the conic section from the point x_1 , i.e., $u_1 = a(w - x_1)$, where a is some scalar, and w is the center of C , which is not known yet.

At this point we will prove the following "conjugacy" conditions:

$$(u_0, Mu_1) = (u_1, Mu_0) = 0,$$

which we will need to determine u_1 .

Theorem: Let C be the conic section defined by Eq. A2.5. Let $u_0 = -r_0$ be the tangent to C at x_1 , and let $u_1 = w - x_1$. Then

$$(u_1, Mu_0) = (u_0, Mu_1) = 0. \quad (A2.6)$$

Proof: As in the case of a hyper-ellipsoid, the center w of C is determined by minimizing

$$F(x) = (1/2) * (x, Mx) - (x, b),$$

the only difference being that now the minimization is done with respect to vectors in the plane P defined by r_0 and $r_1 = \nabla F(x_1)$. To do this minimization we let

$$x = w + (\alpha r_0 + \beta r_1), \quad \alpha \text{ and } \beta \text{ are scalars.}$$

$x \in P$ since w , r_0 , and r_1 are all in the plane P . Substituting for x in Eq. A2.5 yields after some algebra

$$\begin{aligned} F(x) = F(w) + \alpha \{ (r_0, Mw) - (r_0, b) \} \\ + \{ (r_1, Mw) - (r_1, b) \} + \alpha\beta (r_0, Mr_1) \end{aligned} \quad (A2.7)$$

FIG. A2.1. Intersection of the plane P defined by the vectors $\underline{r}_0$ and $\underline{r}_1$, with the hyperellipsoid defined by $F(\underline{x}) = F(\underline{x}_1)$, is the conic section C , with the center w .

$$+ (\alpha^2/2)(r_0, Mr_0) + (\beta^2/2)(r_1, Mr_1)$$

Minimization conditions are

$$\left. \frac{\partial F}{\partial \alpha} \right|_{\alpha} = 0, \quad \left. \frac{\partial F}{\partial \beta} \right|_{\beta} = 0.$$

The first of these yields

$$(r_0, Mw) - (r_0, b) = 0. \quad (A2.8)$$

Recalling that r_0 is tangent to the conic section at the point x_1 , we have

$$(r_0, \nabla F(x_1)) = 0, \text{ or } (r_0, Mx_1 - b) = 0,$$

which gives $(r_0, b) = (r_0, Mx_1)$. Substituting this into Eq. A2.8, we obtain

$$(r_0, Mw) - (r_0, Mx_1) = 0, \text{ or } (r_0, M(w - x_1)) = 0.$$

Since $u_1 = w - x_1$, and $u_0 = -r_0$, we finally get

$$(u_0, Mu_1) = 0$$

$(u_1, Mu_0) = 0$ follows from the self-adjointness of M . Q.E.D.

In practice, since u_1 is coplanar with $u_0 = -r_0$ and $r_1 = \nabla F(x_1)$, we let

$$u_1 = -r_0 + s_1 u_0$$

where s_1 is some scalar that is determined using the conjugacy conditions in Eq. A2.6.

Now that we have our new search direction u_1 , x_2 is chosen along this direction at the point where u_1 is tangent to the hyper-ellipsoid through x_2 , i.e., we let

$$x_2 = x_1 + \lambda_1 u_1$$

and determine λ_1 using the tangency condition

$$(u_1, \nabla F(x_2)) = 0. \quad (A2.9)$$

The general algorithm should be apparent by now. At the end of the i th step, the vectors x_i , r_i , and u_{i-1} are known. Then u_i is determined so that

i) it is coplanar with r_i and u_{i-1} , i.e.,

$$u_i = -r_i + s_i u_{i-1} \quad (A2.10)$$

ii) u_i and u_{i-1} are conjugates,

$$(u_i, Mu_{i-1}) = 0 \quad (A2.11)$$

Then

$$x_{i+1} = x_i + \lambda_i u_i \quad (A2.12)$$

which also gives, after operating with M ,

$$r_{i+1} = r_i + \lambda_i Mu_i. \quad (A2.13)$$

As mentioned earlier, λ_i is determined using the tangency condition

$$(u_i, \nabla F(x_{i+1})) = (u_i, r_{i+1}) = 0 \quad (A2.14)$$

Conjugacy requirement A2.11 determines the scalar s_i . Simple algebra yields

$$\lambda_i = \|r_i\|^2 / (u_i, Mu_i)$$

$$s_i = \|r_i\|^2 / \|r_{i-1}\|^2$$

where $\|r_i\|^2 = (r_i, r_i)$, etc.

In summary, Hestenes-Stiefel conjugate-gradient method proceeds as follows:

Let $r_0 = Mx_0 - b$, and $u_0 = -r_0$. Then

$$\lambda_i = ||r_i||^2 / (u_i, Mu_i) \quad (A2.16a)$$

$$x_{i+1} = x_i + \lambda_i u_i \quad (A2.16b)$$

$$r_{i+1} = r_i + \lambda_i Mu_i \quad (A2.16c)$$

$$s_{i+1} = ||r_{i+1}||^2 / ||r_i||^2 \quad (A2.16d)$$

$$u_{i+1} = -r_{i+1} + s_{i+1} u_i \quad (A2.16e)$$

Gastinel [3] shows that r_i is orthogonal to all r_j , $j < i$, i.e.,

$$(r_i, r_j) = 0 \text{ for } j < i.$$

After n steps, we get the residual $r_n = Mx_n - b$, where now $(r_n, r_j) = 0$ for $j=0,1,2,\dots,n-1$. Since r_j , ($j=0,1,2,\dots,n-1$) are all orthogonal to each other, we may define a basis in terms of them. Thus, an arbitrary vector $v \in R^n$ may be written in terms of this basis as

$$v = \sum_{j=0}^{n-1} \alpha_j r_j$$

Taking the scalar product of r_n with v , we get

$$(r_n, v) = \sum_{j=0}^{n-1} \alpha_j (r_n, r_j) = 0,$$

since $(r_n, r_j) = 0$. Since this is true for any arbitrary vector v , we must have

$$r_n = Mx_n - b = 0.$$

Thus, the iterations terminate after n steps, and x_n is the exact solution.

c) Incomplete Cholesky Conjugate-Gradient (ICCG) Method

Above we discussed incomplete Cholesky decomposition of M , and the conjugate-gradient iterations for the solution of $Mx=b$. Here we will make the connection between these two.

As shown in the previous section, conjugate-gradient method is guaranteed to converge in at most n iterations, where n is the dimension of the system. In practice n is usually a large number, on the order of many hundreds; therefore, if we were to iterate n times, this method would be a very slow one. As Kershaw [4] mentions, the number of iterations needed is indeed large if M has a large spread of eigenvalues, which is the case when M is obtained by finite differencing a partial differential equation. But if M were nearly an identity matrix, its eigenvalues would be clustered around unity, and the iterations would be expected to converge very rapidly. Thus, we try to formulate our problem in terms of an approximate identity matrix.

Since M is symmetric and positive definite, we do an incomplete Cholesky decomposition,

$$M \approx L^t L.$$

Then we write $Mx=y$ as

$$(L^{-1}M(L^t)^{-1})L^tx = L^{-1}y,$$

where $L^{-1}M(L^t)^{-1}$ is an approximate identity matrix, and apply the conjugate-gradient method (A2.16) to the system

$$\tilde{M}x = \tilde{y}$$

where $\tilde{M} = L^{-1}M(L^t)^{-1}$, $\tilde{x} = L^tx$, and $\tilde{y} = L^{-1}y$. Kershaw shows that in this form the method indeed requires very few iterations.

d) Application to Systems with Non-symmetric Matrices

To be able apply these methods to a more general class of systems

$$Ax = y \quad (A2.18)$$

where A is now any non-singular sparse matrix, we do an incomplete LU decomposition,

$$A = LU + E,$$

where E is again some negligible error matrix. Then the problem is put into the form

$$L^{-1}AU^{-1}(Ux) = L^{-1}y, \text{ or } \tilde{A}\tilde{x} = \tilde{y},$$

where $\tilde{A} = L^{-1}AU^{-1}$, $\tilde{x} = Ux$, and $\tilde{y} = L^{-1}y$. Note that $\tilde{A}$ is an approximate identity matrix, but still it is non-symmetric. But the problem can be posed in terms of a symmetric matrix if we multiply on the left by $\tilde{A}^t$:

$$\tilde{A}^t\tilde{A}\tilde{x} = \tilde{A}^t\tilde{y}, \text{ or } M\tilde{x} = z,$$

where $M = \tilde{A}^t\tilde{A}$ is a symmetric matrix, and $z = \tilde{A}^t\tilde{y}$.

Now the original problem has been put in a form to which we can apply the Hestenes-Stiefel conjugate-gradient method as outlined in Eq. A2.16, which yields after some matrix algebra [4],

$$r_0 = Ax_0 - y, \quad u_0 = -(U^tU)^{-1}A^t(LL^t)^{-1}r_0,$$

$$\lambda_i = (r_i, (LL^t)^{-1}r_i) / (u_i, U^tUu_i) \quad (A2.21a)$$

$$x_{i+1} = x_i + \lambda_i u_i \quad (A2.21b)$$

$$r_{i+1} = r_i + \lambda_i Au_i \quad (A2.21c)$$

$$s_{i+1} = (r_{i+1}, (LL^t)^{-1}r_{i+1}) / (r_i, (LL^t)^{-1}r_i) \quad (A2.21d)$$

$$u_{i+1} = -(U^tU)^{-1}A^t(LL^t)^{-1}r_{i+1} + s_{i+1}u_i \quad (A2.21e)$$

$$i=1,2,3,\dots$$

In our code we adopted the subroutine ICLU from the LIBRIS library of MFECC [5], which solves the system $Ax=b$, where A need not be symmetric. ICLU is a highly optimized FORTRAN subroutine which makes use of STACKLIB [6], a vector function library of optimum stack-loops for the CDC 7600.

REFERENCES

- [1] M.R. Hestenes, and E. Stiefel, Nat. Bur. Standards J. Res. 49, 409 (1952)
- [2] G. Dahlquist, and A. Bjorck, "Numerical Methods", Prentice Hall, Englewood Cliffs, N.J. (1970).
- [3] N. Gastinel, "Linear Numerical Analysis", Hermann, Paris (1970).
- [4] D. S. Kershaw, J. Computational Phys. 26, 43 (1978).
- [5] T. Cutler, ICLU, The ICCG Solver, No: 42
in LIBRIS library of Magnetic Fusion Energy Computer Center (MFECC), Lawrence Livermore Laboratory, University of California, Livermore, Ca. (1978).
- [6] F.H. McMahon, L.J. Sloan, and G.A. Long,
LLL Publication UCID 30083, Livermore, Ca. (1976).

```

1 ..... APPENDIX III
2
3
4 ..... COMPUTER PROGRAM LISTING
5
6
7 ..... PARAMETER (IN1=15, IN2=31, IN3=0, JM1P1=26, JM2=15)
8 ..... PARAMETER (NS=IN2+JM1P1, NT=IN1+IN2, MT=JM1P1+JM2, N=(NT-2)+(MT-2))
9 ..... LCM (VARIABLES), (LCMICCG), (CDEFAD), (CONTOUR), (WLCM2)
10 ..... COMMON/VARIABLES/PSI (IN2, JM1P1, 3), B1 (IN2, JM1P1, 2),
11 ..... 1B2 (IN2, JM1P1, 2), B3 (IN2, JM1P1, 2), XI (IN2, JM1P1, 2), C1 (IN2, JM1P1, 2),
12 ..... 1C2 (IN2, JM1P1, 2), C3 (IN2, JM1P1, 2),
13 ..... 2T (IN2, JM1P1, 2), W (4, IN2, JM1P1, 3), F (4, IN2, JM1P1, 3),
14 ..... 3EI (IN2, JM1P1, 2), CN (IN2, JM1P1, 2)
15 ..... COMMON/CONST2/IS, IB, IE, IB, IP, JS, JB, JE, JB, JP, BX, BY, BATA, BATB
16 ..... COMMON/MESH/RR (NT), ZZ (MT), DELRR (NT-1), DELZZ (MT-1),
17 ..... 1SR (IN2), SZ (JM1P1), DELR (IN2-1), DELZ (JM1P1-1)
18 ..... COMMON/CROUTY/AA (IN2, 3), D (IN2), X1 (IN2)
19 ..... COMMON/CONST/N1, N2, N3, M1, M2, N1P1, N1P2, N2P1, N2P2, N2M1,
20 ..... 1N1PN2, N1PN2P, N1PN2M, M1P1, M1M1, M1P2, M2P1, M2M1, M2P2, NN, NNM1, MM, MMM1
21 ..... COMMON/CUR/NCUR, LDCUR (35), VALCUR (35)
22 ..... COMMON/DIM/RMA, RB, BB1, RLENGTH, ZLENGTH
23 ..... COMMON/CDEFAD/CDEFR1 (IN2-1, JM1P1-1), CDEFR2 (IN2-1, JM1P1-1),
24 ..... 2CDEFR3 (IN2-1, JM1P1-1), CDEFZ1 (IN2-1, JM1P1-1),
25 ..... 3CDEFZ2 (IN2-1, JM1P1-1), CDEFZ3 (IN2-1, JM1P1-1)
26 ..... COMMON/CONTOUR/FLUXRZ (NT, MT), RAXIS (NT), ZAXIS (MT)
27 ..... COMMON/VAR2/FR, FK, CY, C, Q, HU, UN, BT, TMAX, PTIME1, PTIME2, TIME,
28 ..... 1TBIS1, TBIS2,
29 ..... 1BMIN, SGO, COE, MW, MP, MC, NCURV, NCHECK, BTOR, BN, TEMP,
30 ..... 2CUR1MAX, CUR2MAX, CUR3MAX, CURTOR, RSPINCH, RBPINCH
31 ..... COMMON/WSCM/S (N, (-4, 4))
32 ..... COMMON/WLCM2/GA (IN2, JM1P1), GB (IN2, JM1P1), GC (IN2, JM1P1),
33 ..... 1GB (IN2, JM1P1), GE (IN2, JM1P1), GF (IN2, JM1P1), YSCM1 (4, IN2, JM1P1),
34 ..... 2YSCM2 (4, IN2, JM1P1), APSI (NT, MT)
35 ..... COMMON/LCMICCG/MTRX (N, (-4, 4)), LU (N, (-4, 4)), X (N), Y (N)
36 ..... REAL MTRX, LU
37 C ..... MW ..... OF STEPS BETWEEN WRITES
38 C ..... FR ..... GAS CONSTANT
39 C ..... CY ..... SPECIFIC HEAT AT CONSTANT VOLUME
40 C ..... C ..... SPEED OF LIGHT

```

```

41 C ..... FK ..... GAMMA OF GAS, 5/3
42 C ..... BMIN ..... TOROIDAL BIAS FIELD
43 C ..... BTOR ..... MAXIMUM TOROIDAL FIELD
44 C ..... RB ..... MAJOR RADIUS
45 C ..... RSMA ..... MINOR RADIUS
46 C ..... DN ..... FILLING DENSITY
47 C ..... PTIME= ..... RISE TIME OF THE FIELDS
48 C ..... SGO ..... INITIAL ELECTRICAL CONDUCTIVITY
49 C ..... B1, B2, B3 ..... BR, BZ, BPHI (MAGNETIC FIELD COMPONENTS)
50 C ..... C1, C2, C3 ..... JR, JZ, JPHI (CURRENT COMPONENTS)
51 C ..... W(1, I, J, K) ... RHO ..... (DENSITY)
52 C ..... W(2, I, J, K) ... RHO*UR
53 C ..... W(3, I, J, K) ... RHO*UZ
54 C ..... W(4, I, J, K) ... RHO*UPHI
55 C ..... UR, UZ, UPHI ... COMPONENTS OF VELOCITY U
56 C ..... IN CYLINDRICAL COORDINATES
57 C ..... SR ..... R (RADIUS)
58 C ..... T ..... ELECTRICAL CONDUCTIVITY
59 C ..... CN ..... HEAT CONDUCTIVITY
60 C ..... PSI ..... POLOIDAL FLUX
61 C ..... XI ..... TOROIDAL FLUX
62 C ..... EI ..... TEMPERATURE
63 C ..... MP ..... OF STEPS BETWEEN PLOTS
64 C ..... MC ..... OF STEPS BETWEEN CONTOUR PLOTS
65 C ..... NCURV ..... OF CONTOURS IN A PLOT
66 C ..... TEMP ..... INITIAL TEMPERATURE (KELVIN)
67 ..... OPTIMIZE
68 ..... DATA C1R1, C1R2, C2Z1, C2Z2, C3Z2 / 10.75, 18.25, 5.25, 10.75, 3.25 /
69 ..... DATA RSMA, RB, BB1, RLENGTH, ZLENGTH / 7.5, 15.0, 12., 1., 1. /
70 ..... CALL DROFFILE(0)
71 ..... CALL OPEN(2, "INPUT", 0, LEN)
72 ..... CALL CREATE(3, "OUT", 3, -1)
73 C
74 C ..... READ IN DATA DESCRIBING THE RUN
75 C
76 ..... READ(2, 2) BMIN, BTOR, DN, TEMP
77 ..... READ(2, 2) BT, TMAX, TBIS1, TBIS2, PTIME1, PTIME2, CONBUC
78 ..... READ(2, 2) CUR1MAX, CUR2MAX, CUR3MAX, CURTOR
79 ..... READ(2, 3) MW, MP, MC, NCURV, NCHECK
80 ..... READ(2, 2) RSPINCH, RBPINCH, RZERO, PNCH1, TAU0, TAU1

```

```

81.....2 FORMAT (7E10.2)
82.....3 FORMAT (5I5)
83.....10 FORMAT (-7F6.3)
84.....N1=IN1
85.....N2=IN2
86.....N3=IN3
87.....M1=JM1P1-1
88.....M2=JM2
89.....C1R1=C1R1/RSMA...$...C1R2=C1R2/RSMA
90.....C2Z1=C2Z1/RSMA...$...C2Z2=C2Z2/RSMA
91.....C3Z2=C3Z2/RSMA
92.....BT=BT/1.0E+6
93.....TMAX=TMAX/1.0E+6
94.....TDIS1=TDIS1/1.0E+6
95.....TDIS2=TDIS2/1.0E+6
96.....PTIME1=PTIME1/1.0E+6
97.....PTIME2=PTIME2/1.0E+6
98.....TAU0=TAU0/1.0E+6
99.....TAU1=TAU1/1.0E+6
100.....CUR1MAX=CUR1MAX*3.0E+9
101.....CUR2MAX=CUR2MAX*3.0E+9
102.....CUR3MAX=CUR3MAX*3.0E+9
103.....CURTOR=CURTOR*3.0E+9
104.....FR=2.0/2.0*8.31E+7
105.....FK=5./3.
106.....CV=FR/(FK-1.)
107.....C=2.9980E+10
108.....Q=RSMA/RB
109.....HU=0.10
110.....SMALL=1.0E-30
111.....TMALL=SMALL**2
112...BRK CONTINUE
113.....WRITE (3,1) BMIN,BTOR,CONBUC,BT,TMAX,PTIME1,PTIME2,TDIS1,TDIS2,
114.....1TAU0,TAU1,BN,TEMP,CUR1MAX,CUR2MAX,CUR3MAX,CURTOR
115.....1,RSPINCH,RBPINCH,RZERO,PNGHT
116.....1 FORMAT (/5X,"BMIN...=",E11.3," GAUSS."
117.....1/5X,"BTOR...=",E11.3," GAUSS."
118.....2/5X,"CONBUC...=",E11.3," /SEC."
119.....2/5X,"BT...=",E11.3," SEC."
120.....3/5X,"TMAX...=",E11.3," SEC."
121.....4/5X,"PTIME1...=",E11.3," SEC."

```

```

122 ..... 5/5X, "PTIME2 =" , E11.3, " SEC."
123 ..... 6/5X, "TBIS1 .. =" , E11.3, " SEC."
124 ..... 7/5X, "TBIS2 .. =" , E11.3, " SEC."
125 ..... 1/5X, "TAU0 .... =" , E11.3, " SEC."
126 ..... 2/5X, "TAU1 .... =" , E11.3, " SEC."
127 ..... 8/5X, "BN ..... =" , E11.3, " GM/CC"
128 ..... 9/5X, "TEMP .... =" , E11.3, " DEG. KELVIN"
129 ..... 2/5X, "CUR1MAX =" , E11.3, " STATAMPS."
130 ..... 1/5X, "CUR2MAX =" , E11.3, " STATAMPS."
131 ..... 2/5X, "CUR3MAX =" , E11.3, " STATAMPS."
132 ..... 3/5X, "CURTOR =" , E11.3, " STATAMPS."
133 ..... 2/5X, "RSPINCH =" , E11.3, " CM."
134 ..... 3/5X, "RBPINCH =" , E11.3, " CM."
135 ..... 4/5X, "RZERO .. =" , E11.3, " CM."
136 ..... 5/5X, "PNCHT .. =" , E11.3, " EV."
137 C
138 C ..... DO SOME INITIALIZATION
139 C
140 ..... CALL INITIAL (CONDOC, RZERO, PNCHT, TAU0, TAU1, IC1B, IC1E, JC2B, JC2E,
141 ..... 1JC3E, C1R1, C1R2, C2Z1, C2Z2, C3Z2)
142 ..... CALL ICLU (LUERR)
143 ..... WRITE (3,13) LUERR
144 ..... 13 FORMAT (/5X, "LUERR =", I6/)
145 ..... 5 TIME=0.
146 ..... M=0
147 ..... ABDEEM=0.
148 ..... ISWT=0
149 C ..... START OF MAIN LOOP
150 ..... 6 M=M+1
151 ..... ISWCH=0
152 ..... K=1
153 ..... TIME=TIME+DT
154 C
155 C ..... DO BOUNDARY CONDITIONS ON XI, AND UPDATE THE EXTERNAL CURRENTS
156 C
157 ..... TAU1=PTIME2-TAU0
158 ..... IF (TIME.GE.PTIME1) GO TO 400
159 ..... DO 8 I=IS, ID
160 ..... XI (I, JB, K+1) = ((1.-BMIN) ♦ SIN(3.14159 ♦ TIME

```

```

161.....1/(2.*PTIME1))+BMIN)/Q
162.....8*CONTINUE
163.....DO 9 J=JS,JB
164.....XI (IB,J,K+1)=XI (IB,JB,K+1)
165.....XI (IS,J,K+1)=XI (IS,JB,K+1)
166.....9*CONTINUE
167...400 IF (TIME.GT.PTIME2) GO TO 7
168.....L=0
169.....DO 401 I=IC1B,IC1E
170.....L=L+1
171.....VALCUR(L)=+CUR1MAX*SIN(3.14159*(TIME-TAU0)/(2.*TAU1))
172.....IF (TIME.LT.TAU0) VALCUR(L)=0.0
173...401 CONTINUE
174.....DO 402 J=JC2B,JC2E
175.....L=L+1
176...402 VALCUR(L)=+CUR2MAX*SIN(3.14159*TIME/(2.*PTIME2))
177.....DO 403 J=2,JC3E
178.....L=L+1
179...403 VALCUR(L)=+CUR3MAX*SIN(3.14159*TIME/(2.*PTIME2))
180.....7*CONTINUE
181.....K=1
182...C.....CALCULATE THE PREDICTED VALUE OF W
183.....CALL QACOPY(GA,B1(1,1,1),NS)
184.....CALL QACOPY(GB,B2(1,1,1),NS)
185.....CALL QACOPY(GC,B3(1,1,1),NS)
186.....CALL QACOPY(GD,B1(1,1,1),NS)
187.....CALL QACOPY(GE,B2(1,1,1),NS)
188.....CALL QACOPY(GF,B3(1,1,1),NS)
189.....CALL QACOPY(VSEM1,W(1,1,1,1),4*NS)
190.....DO 433 J=JB,JE
191.....DO 434 I=IB,IE
192.....F(1,I,J,K)=0.
193.....VSEM2(1,I,J)=0.25*(VSEM1(1,I+1,J)+VSEM1(1,I-1,J)+VSEM1(1,I,J+1)+
194.....1VSEM1(1,I,J-1))-0.50*BT/BX/SR(I)*(SR(I+1)*VSEM1(2,I+1,J)-
195.....2SR(I-1)*VSEM1(2,I-1,J))-0.50*BT/BY*(VSEM1(3,I,J+1)-VSEM1(3,I,J-1))
196.....F(2,I,J,K)=(VSEM1(4,I,J)+2*(VSEM1(1,I,J)*SR(I)+(FK-1.)*VSEM1(1,I
197.....9,J))
198.....1EI(1,I,J,K)/SR(I)+(GF(1,J)*GB(1,J)-GE(1,J)*GD(1,J))*BT/2.
199.....VSEM2(2,I,J)=0.25*(VSEM1(2,I+1,J)+VSEM1(2,I-1,J)+VSEM1(2,I,J+1)+
200.....1VSEM1(2,I,J-1))-0.50*BT/BX/SR(I)*(SR(I+1)*VSEM1(2,I+1,J)+

```



```

201 ..... 2/VSEM1(1,I+1,J)+(FK-1.)*VSEM1(1,I+1,J)*EI(1,I+1,J,K))-
202 ..... 3SR(I-1)*(VSEM1(2,I-1,J)**2/VSEM1(1,I-1,J)+(FK-1.)*VSEM1(1,I-1,J)
203 ..... 4*EI(1,I-1,J,K)))-0.50*BT/BY*(VSEM1(2,I,J+1)+VSEM1(3,I,J+1)
204 ..... 5/VSEM1(1,I,J+1)-VSEM1(2,I,J-1)+VSEM1(3,I,J-1)/VSEM1(1,I,J-1))
205 ..... 6+2.*F(2,I,J,K)
206 ..... F(3,I,J,K)=(GB(1,J)+GC(1,J)-GF(1,J)+GA(1,J))*BT/2.
207 ..... VSEM2(3,I,J)=0.25*(VSEM1(3,I+1,J)+VSEM1(3,I-1,J)+VSEM1(3,I,J+1)+
208 ..... 1VSEM1(3,I,J-1))-0.50*BT/BX/SR(I)*(SR(I+1)+VSEM1(2,I+1,J)+
209 ..... 2VSEM1(3,I+1,J)/VSEM1(1,I+1,J)-SR(I-1)+VSEM1(2,I-1,J)+VSEM1(3,I-1,J
210 ..... 9)
211 ..... 3/VSEM1(1,I-1,J))-0.50*BT/BY*((VSEM1(3,I,J+1)**2/VSEM1(1,I,J+1)+
212 ..... 4*(FK-1.)*VSEM1(1,I,J+1)*EI(1,I,J+1,K))-(VSEM1(3,I,J-1)**2/
213 ..... 5VSEM1(1,I,J-1)+(FK-1.)*VSEM1(1,I,J-1)*EI(1,I,J-1,K))
214 ..... 6+2.*F(3,I,J,K)
215 ..... F(4,I,J,K)=(GE(1,J)+GA(1,J)-GB(1,J)+GB(1,J))
216 ..... 1-VSEM1(2,I,J)+VSEM1(4,I,J)/(VSEM1(1,I,J)+SR(I))*BT/2.
217 ..... VSEM2(4,I,J)=0.25*(VSEM1(4,I+1,J)+VSEM1(4,I-1,J)+VSEM1(4,I,J+1)+
218 ..... 1VSEM1(4,I,J-1))-0.50*BT/BX/SR(I)*(SR(I+1)+VSEM1(2,I+1,J)
219 ..... 2*VSEM1(4,I+1,J)/VSEM1(1,I+1,J)-SR(I-1)+VSEM1(2,I-1,J)+VSEM1(4,I-1,
220 ..... 9J)
221 ..... 3/VSEM1(1,I-1,J))-0.50*BT/BY*(VSEM1(3,I,J+1)+VSEM1(4,I,J+1)
222 ..... 4/VSEM1(1,I,J+1)-VSEM1(3,I,J-1)+VSEM1(4,I,J-1)/VSEM1(1,I,J-1))
223 ..... 5+2.*F(4,I,J,K)
224 ..... DD 435 L=1,4
225 ..... IF (ABS(VSEM2(L,I,J)).L1.SMALL) VSEM2(L,I,J)=0.
226 ..... 435 CONTINUE
227 ..... 434 CONTINUE
228 C..... SINCE W(2,I,J,K) IS NOT EQUAL TO ZERO ANYMORE ON THE SIDE WALLS
229 C..... WE HAVE TO CORRECT VSEM2(2,,)
230 ..... VSEM2(2,IE,J)=VSEM2(2,IE,J)+2.0*BT/(2.0*SR(IE)+BX)
231 ..... 1*(SR(IE+1)+VSEM1(2,IE+1,J)**2/VSEM1(1,IE+1,J))
232 ..... VSEM2(2,IB,J)=VSEM2(2,IB,J)-2.0*BT/(2.0*SR(IB)+BX)
233 ..... 1*(SR(IB-1)+VSEM1(2,IB-1,J)**2/VSEM1(1,IB-1,J))
234 ..... 433 CONTINUE
235 ..... DD 438 I=IB,IE
236 ..... 438 VSEM2(3,I,JE)=VSEM2(3,I,JE)+
237 ..... 1-2.0*0.5*BT/BY*(VSEM1(3,I,JE+1)**2/VSEM1(1,I,JE+1))
238 C..... DD BC'S ON PREDICTED W
239 ..... DD 208 J=JS,JB
240 ..... DD 209 L=1,4

```

```

241.....VSEM2(L,IB,J)=VSEM2(L,IE,J)
242.....VSEM2(L,IS,J)=VSEM2(L,IB,J)
243...209·CONTINUE
244...208·CONTINUE
245·C
246.....DO·300·I=IS,IB
247.....DO·301·L=1,4
248.....VSEM2(L,I,JB)=VSEM2(L,I,JE)
249.....VSEM2(L,I,JS)=VSEM2(L,I,JB+1)
250...301·CONTINUE
251.....VSEM2(3,I,JS)=-VSEM2(3,I,JB+1)
252.....VSEM2(4,I,JS)=-VSEM2(4,I,JB+1)
253...300·CONTINUE
254.....CALL·QACOPY(W(1,1,1,2),VSEM2,4*NS)
255·C
256·C.....CALCULATE·PSI(N+1),XI(N+1),T(N+1)
257·C.....BUT·FIRST·CHANGE·THE·BOUNDARY·VALUES
258...440·DO·302·L=1,2
259.....DO·123·J=JS,JE
260.....W(2,IS,J,L)=0.0
261...123·W(2,IB,J,L)=0.0
262.....DO·124·I=IS,IB
263...124·W(3,I,JB,L)=0.0
264...302·CONTINUE
265·C.....CALCULATE·PSI
266.....CALL·FLX(M,LERR)
267.....IF(LERR.EQ.0)·GO·TO·388
268·C.....ITERATIONS·FOR·PSI·DID·NOT·CONVERGE;·STOP·HERE
269.....TMAX=-TIME·$·MM=M-1
270.....MP=M-1·$·MC=M-1
271.....GO·TO·256
272...388·CONTINUE
273·C.....CALCULATE·XI·AND·TEMPERATURE
274...391·CALL·ADI(M)
275·C.....CALCULATE·THE·NEW·ELECTRICAL·CONDUCTIVITY.
276.....K=1
277...392·DO·86·J=JB,JE
278.....DO·86·I=IB,IE
279...117·IF·(EI(I,J,K+1).GT.0)·GO·TO·115
280.....WRITE(3,118)·I,J,M,BT,TIME,EI(I,J,K+1)

```

```

281 ..... EI(I,J,K+1)=0.25*(EI(I+1,J,K+1)+EI(I-1,J,K+1)+
282 ..... 1 ..... EI(I,J+1,K+1)+EI(I,J-1,K+1))
283 ..... ISWT=ISWT+1
284 ..... IF (ISWT.LE.3) GO TO 117
285 ..... TMAX=-TIME ..$.. MW=M-1 ..$.. MP=M-1 ..$.. MC=M-1
286 ..... GO TO 253
287 ... 115 T(I,J,K+1)=(CONDUCT*(UN*UN*EI(I,J,K+1)/CV/11600.)*♦♦1.5)/SGD
288 ... 118 FORMAT(1H0,3I6,3E16.7/5X,"NEGATIVE TEMPERATURE")
289 ... 86 CONTINUE
290 ..... DO 87 I=IB,IE
291 ..... T(I,JS,K+1)=T(I,JB+1,K+1)
292 ... 87 CONTINUE
293 ..... K2=2
294 ..... DO 33 J=JB,JE
295 ..... DO 34 I=IB,IE
296 ..... F(1,I,J,K2)=0.
297 ..... F(2,I,J,K2)=(C3(I,J,K2)*B2(I,J,K2)-C2(I,J,K2)*B3(I,J,K2))*BT/2.
298 ..... F(3,I,J,K2)=(C1(I,J,K2)*B3(I,J,K2)-C3(I,J,K2)*B1(I,J,K2))*BT/2.
299 ..... F(4,I,J,K2)=(C2(I,J,K2)*B1(I,J,K2)-C1(I,J,K2)*B2(I,J,K2))*BT/2.
300 ... 34 CONTINUE
301 ... 33 CONTINUE
302 C:.... ADJUST THE B.C. (S FOR XYIMCRD...)
303 ..... DO 120 L=1,2
304 ..... DO 121 J=JS,JE
305 ..... W(2,IS,J,L)=-(ABS(W(2,IB,J,L))+W(2,IB,J,L))/2.
306 ... 121 W(2,IE,J,L)=(ABS(W(2,IE,J,L))-W(2,IE,J,L))/2.
307 ..... DO 122 I=IS,IE
308 ... 122 W(3,I,JB,L)=(ABS(W(3,I,JE,L))-W(3,I,JE,L))/2.
309 ... 120 CONTINUE
310 C:.... CALCULATE MASS AND MOMENTUM DENSITIES AGAIN
311 ..... IF (MOD(M,2).NE.1) GO TO 52
312 ..... CALL XYIMCRD
313 ..... GO TO 253
314 ... 52 CALL XYIMCRD
315 ... 253 CONTINUE
316 ..... K=1
317 ..... CALL QACOPY(W(1,1,1,K),W(1,1,1,3),4*NS)
318 ..... DO 96 I=IB,IE
319 ..... DO 96 J=JB,JE
320 ..... DO 98 L=1,4
321 ..... IF (ABS(W(L,I,J,K)).LT.SMALL) W(L,I,J,K)=0.

```

```

322....98 CONTINUE
323.....IF (W(1,I,J,K) .GT. HU) GO TO 96
324.....XR=1.0
325.....IF (J.EQ.2) XR=0.5
326.....ABEDM=ABEDM+XR*SR(I) + (HU-W(1,I,J,K))
327.....W(1,I,J,K)=HU
328....96 CONTINUE
329.C....DO BC'S FOR W ON THE WALLS
330.C
331....441 DO 28 J=JS,JE
332.....DO 29 L=1,4
333.....W(L,IB,J,K)=W(L,IE,J,K)
334.....W(L,IS,J,K)=W(L,IB,J,K)
335....29 CONTINUE
336.....W(2,IB,J,K)=+(ABS(W(2,IE,J,K))-W(2,IE,J,K))/2.
337.....W(2,IS,J,K)=- (ABS(W(2,IB,J,K))+W(2,IB,J,K))/2.
338....28 CONTINUE
339.C
340.....DO 53 I=IS,IB
341.....DO 54 L=1,4
342.....W(L,I,JB,K)=W(L,I,JE,K)
343.....W(L,I,JS,K)=W(L,I,JB+1,K)
344....54 CONTINUE
345.....W(3,I,JB,K)=+(ABS(W(3,I,JE,K))-W(3,I,JE,K))/2.
346.....W(3,I,JS,K)=-W(3,I,JB+1,K)
347.....W(4,I,JS,K)=-W(4,I,JB+1,K)
348....53 CONTINUE
349.C
350.....IF (MOD(M,MW) .NE. 1 .AND. MOD(M,MP) .NE. 1 .AND. MOD(M,MC) .NE. 1) GO TO 91
351....256 ISWCH=1
352.....SIME=TIME+RSMA/UN
353.....SBT=BT+RSMA/UN
354.....WRITE (3,104) M,SIME,BT,SBT,CDE
355....104 FORMAT(1H1," CYCLE=",I5,5X,"TIME=",E14.7,5X,"BT=",E14.7,
356.....1,"(",E14.7," SEC.)",5X,"CDE=",E14.7,/)
357.....WRITE (3,105)
358....105 FORMAT(" I J POL.FLUX TOR.FLUX RAD.MAGN.",
359.....1" Z-MAGNET TOR.MAGN RAD.CURR Z-CURRENT",
360.....2" TOR.CURR J*B/C (R,Z,TOR.)"/)

```

```

361.....KNT=4
362.....K=1
363.....BD·92·I=IS,IB
364.....BD·92·J=JS,JB
365.....XI·(I,J,K)=XI·(I,J,K+1)·BTOR·RSMA
366.....PSI(I,J,K)=PSI(I,J,K+1)·BTOR·RSMA·2
367.....B1·(I,J,K)=B1·(I,J,K+1)·BTOR
368.....B2·(I,J,K)=B2·(I,J,K+1)·BTOR
369.....B3·(I,J,K)=B3·(I,J,K+1)·BTOR
370.....C1·(I,J,K)=C1·(I,J,K+1)·C·BTOR/(4·3·1416·RSMA)
371.....C2·(I,J,K)=C2·(I,J,K+1)·C·BTOR/(4·3·1416·RSMA)
372.....C3·(I,J,K)=C3·(I,J,K+1)·C·BTOR/(4·3·1416·RSMA)
373.....XR=(C3(I,J,K)·B2(I,J,K)-C2(I,J,K)·B3(I,J,K))/C
374.....XZ=(C1(I,J,K)·B3(I,J,K)-C3(I,J,K)·B1(I,J,K))/C
375.....XP=(C2(I,J,K)·B1(I,J,K)-C1(I,J,K)·B2(I,J,K))/C
376.....WRITE(3,106)·I,J,PSI(I,J,K),XI(I,J,K),B1(I,J,K),
377.....B2(I,J,K),B3(I,J,K),C1(I,J,K),C2(I,J,K),
378.....C3(I,J,K),XR,XZ,XP
379.....KNT=KNT+1
380.....IF(KNT·LT·59)·GO·TO·92
381.....KNT=2
382.....WRITE(3,110)
383.....WRITE(3,105)
384...110·FORMAT(1H1)
385...92·CONTINUE
386.....WRITE(3,112)
387...112·FORMAT(//5X,"CUSP·CURRENTS"/)
388.....BD·113·L=1,NEUR
389.....XA=VALCUR(L)·C·BTOR/(4·3·14159·RSMA)/3.0E+9
390...113·WRITE(3,114)·LOCUR(L),XA
391...114·FORMAT(5X,"LOCATION=",I5,5X,"VALUE=",E11.3,"·AMPS./CM/CM")
392...106·FORMAT(2I3,5E11.3,3E10.2,3E9.1)
393.....WRITE(3,107)
394...107·FORMAT(1H0,2X,"I",2X,"J",5X,"DENSITY",6X,"RADIAL·VELOCITY",2X,
395.....1"Z--VELOCITY",3X,"TORDID·VELOCITY",1X,"INTERNAL·ENERGY",2X,
396.....2"CONDUCTIVITY",4X,"TEMPERATURE")
397·C.....CALCULATE·VELOCITIES
398.....BD·93·I=IS,IB
399.....BD·93·J=JB,JB
400.....F(2,I,J,K+1)=W(2,I,J,K)/W(1,I,J,K)·UN

```

```

401.....F(3,I,J,K+1)=W(3,I,J,K)/W(1,I,J,K)*UN
402.....F(4,I,J,K+1)=W(4,I,J,K)/W(1,I,J,K)*UN
403.....F(1,I,J,K+1)=EI(1,I,J,K+1)/CV*UN*UN
404.....T(1,I,J,K)=T(1,I,J,K+1)*SGO
405.....WRITE(3,108) I,J,W(1,I,J,K),F(2,I,J,K+1),F(3,I,J,K+1),
406.....1F(4,I,J,K+1),EI(1,I,J,K+1),T(1,I,J,K),F(1,I,J,K+1)
407.....93 CONTINUE
408.....108 FORMAT(1H,2I3,7(E14.3,2X))
409.....TOTALM=0.
410.....DO 99 I=IB,IE
411.....DO 99 J=JB,JE
412.....XR=1.0
413.....IF(J.EQ.JB) XR=0.5
414.....99 TOTALM=TOTALM+XR*W(1,I,J,1)*SR(I)
415.....TOTALM=2.0*TOTALM*BN*BX*BY*2.0*3.14159*RSMA**3
416.....ADDEDM2=2.0*ADDEDM*BN*BX*BY*2.0*3.14159*RSMA**3
417.....WRITE(3,100) TOTALM,ADDEDM2
418.....100 FORMAT(///5X,"TOTAL MASS=",E10.2," GM."
419.....1/5X,"ADDED MASS=",E10.2," GM.")
420.....IF(MOD(M,MP).NE.1) GO TO 610
421.....CALL CLPLOT(0,0,1,2,1,2,M,RSMA,UN,TIME,NCURY,BTOR)
422.....610 IF(MOD(M,MC).NE.1) GO TO 91
423.....CALL CLPLOT(1,1,0,0,0,0,M,RSMA,UN,TIME,NCURY,BTOR)
424.....91 CONTINUE
425.....UPDATE VARIABLES
426.....CALL QACOPY(T(1,1,1),T(1,1,2),NS)
427.....CALL QACOPY(EI(1,1,1),EI(1,1,2),NS)
428.....CALL QACOPY(PSI(1,1,1),PSI(1,1,2),NS)
429.....CALL QACOPY(XI(1,1,1),XI(1,1,2),NS)
430.....CALL QACOPY(B1(1,1,1),B1(1,1,2),NS)
431.....CALL QACOPY(B2(1,1,1),B2(1,1,2),NS)
432.....CALL QACOPY(B3(1,1,1),B3(1,1,2),NS)
433.....CALL QACOPY(G1(1,1,1),G1(1,1,2),NS)
434.....CALL QACOPY(G2(1,1,1),G2(1,1,2),NS)
435.....CALL QACOPY(G3(1,1,1),G3(1,1,2),NS)
436.....IF(TIME.LE.TMAX) GO TO 6
437.....IF(ISWCH.EQ.1) GO TO 94
438.....MP=M-1
439.....MC=M-1
440.....GO TO 256

```

```

441 C.....END OF MAIN LOOP
442 .....94 CONTINUE
443 .....WRITE (3,109) SIME, SBT, M, MW, MP, MC, NCHECK, ISWT, LERR
444 .....109 FORMAT (//5X, "***** END OF THE RUN. *****"//
445 .....15X, "TIME ..=", E12.3/5X, "DT .....", E12.3/5X, "CYCLE ..=", 15/
446 .....25X, "MW .....", 15...../5X, "MP .....", 15...../5X, "MC .....", 15/
447 .....35X, "NCHECK=", 15/5X, "ISWT ..=", 15/5X, "LERR ..=", 15)
448 .....CALL EXIT
449 .....END
450
451
452
453 .....SUBROUTINE ADI (M)
454 C..... THIS SUBROUTINE CALCULATES TEMPERATURE
455 C..... AND THE FUNCTION XI USING AN ADI SCHEME
456 .....LCM (VARIABLES), (LCMICCG), (COEFAD), (CONTOUR), (WLCM2)
457 .....COMMON/VARIABLES/PSI (IN2, JM1P1, 3), B1 (IN2, JM1P1, 2),
458 .....1B2 (IN2, JM1P1, 2), B3 (IN2, JM1P1, 2), XI (IN2, JM1P1, 2), C1 (IN2, JM1P1, 2),
459 .....1C2 (IN2, JM1P1, 2), C3 (IN2, JM1P1, 2),
460 .....2T (IN2, JM1P1, 2), W (4, IN2, JM1P1, 3), F (4, IN2, JM1P1, 3),
461 .....3EI (IN2, JM1P1, 2), CN (IN2, JM1P1, 2)
462 .....COMMON/CONST2/IS, IB, IE, ID, IP, JS, JB, JE, JD, JP, IX, IY, BATA, BATB
463 .....COMMON /MESH/ RR (NT), ZZ (MT), DELRR (NT-1), DELZZ (MT-1),
464 .....1SR (IN2), SZ (JM1P1), DELR (IN2-1), DELZ (JM1P1-1)
465 .....COMMON/CROUTY/ AA (IN2, 3), B (IN2), X1 (IN2)
466 .....COMMON/CONST/ N1, N2, N3, M1, M2, N1P1, N1P2, N2P1, N2P2, N2M1,
467 .....1N1PN2, N1PN2P, N1PN2M, M1P1, M1M1, M1P2, M2P1, M2M1, M2P2, NN, NNM1, MM, MMM1
468 .....COMMON/CUR/ NCUR, LOGCUR (35), VALCUR (35)
469 .....COMMON/DIM/RMA, RB, BB1, RLNGTH, ZLNGTH
470 .....COMMON/COEFAD/COEFR1 (IN2-1, JM1P1-1), COEFR2 (IN2-1, JM1P1-1),
471 .....2COEFR3 (IN2-1, JM1P1-1), COEFZ1 (IN2-1, JM1P1-1),
472 .....3COEFZ2 (IN2-1, JM1P1-1), COEFZ3 (IN2-1, JM1P1-1)
473 .....COMMON/CONTOUR/ FLUXR2 (NT, MT), RAXIS (NT), ZAXIS (MT)
474 .....COMMON/VAR2/FR, FK, CY, C, Q, HU, UN, DT, TMAX, PTIME1, PTIME2, TIME,
475 .....1 TBIS1, TBIS2,
476 .....1BMIN, SGB, COE, MW, MP, MC, NCURY, NCHECK, BTOR, BN, TEMP,
477 .....2CUR1MAX, CUR2MAX, CUR3MAX, CURTOR, RSPINCH, RBPINCH
478 .....COMMON /WSEM/ S (N, (-4, 4))
479 .....COMMON/WLCM2/ GA (IN2, JM1P1), GB (IN2, JM1P1), GC (IN2, JM1P1),
480 .....1GB (IN2, JM1P1), GE (IN2, JM1P1), GF (IN2, JM1P1), VSEM1 (4, IN2, JM1P1),

```

```

481 ..... 2VSEM2(4,IN2,JM1P1),APSI(NT,MT)
482 ..... COMMON /LEM1666/ MTRX(N,(-4,4)),LU(N,(-4,4)),X(N),Y(N)
483 ..... REAL MTRX,LU
484 ..... OPTIMIZE
485 ..... K=1
486 ..... CALL QACOPY(VSEM1,W(1,1,1,1),4*NS)
487 ..... CALL QACOPY(VSEM2,W(1,1,1,2),4*NS)
488 ..... CALL QACOPY(GE,T(1,1,2),NS)
489 ..... CALL QACOPY(GF,XI(1,1,1),NS)
490 ... 405 IF (MOD(M,2).NE.1) GO TO 312
491 ... 311 CONTINUE
492 ..... DO 16 J=JB,JE
493 ..... DO 15 I=IB,IE
494 ..... GC(I,J)=CDE/(BY*BY*(GE(I,J-1)+GE(I,J))/2.)
495 ..... 1+BATB*VSEM2(3,I,J)/VSEM2(1,I,J)
496 ..... GA(I,J)=CDE/(BY*BY*(GE(I,J+1)+GE(I,J))/2.)
497 ..... 1-BATB*VSEM2(3,I,J)/VSEM2(1,I,J)
498 ..... GB(I,J)=1.+CDE/(BY*BY*(
499 ..... 1*(2./(GE(I,J+1)+GE(I,J))
500 ..... 2+2./(GE(I,J-1)+GE(I,J))
501 ..... 3+BT/2.0*(SR(I)/(2.0*BK)*
502 ..... 4(VSEM2(2,I+1,J)/VSEM2(1,I+1,J)/SR(I+1)
503 ..... 5-VSEM2(2,I-1,J)/VSEM2(1,I-1,J)/SR(I-1))
504 ..... 6+(VSEM2(3,I,J+1)/VSEM2(1,I,J+1)-VSEM2(3,I,J-1)/VSEM2(1,I,J-1))
505 ..... 1/(2.0*BY))
506 ..... GB(I,J)=GF(I,J)+CDE*(SR(I)/(BK*BK)*
507 ..... 1(GF(I+1,J)-GF(I,J))/(GE(I+1,J)+GE(I,J))/2.
508 ..... 2*(SR(I+1)+SR(I))/2.)-(GF(I,J)-GF(I-1,J))
509 ..... 3/(GE(I-1,J)+GE(I,J))/2.+(SR(I-1)+SR(I))/2.))
510 ..... 3-BT/2.0*GF(I,J)*(SR(I)/(2.0*BK)*
511 ..... 4(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)/SR(I+1)
512 ..... 5-VSEM1(2,I-1,J)/VSEM1(1,I-1,J)/SR(I-1))
513 ..... 6+(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))
514 ..... 1/(2.0*BY))
515 ..... 6+BT*SR(I)*
516 ..... 7*(B1(I+1,J,K)*VSEM1(4,I+1,J)/VSEM1(1,I+1,J)
517 ..... 8*-B1(I-1,J,K)*VSEM1(4,I-1,J)/VSEM1(1,I-1,J))/(2.*BK)
518 ..... 9+(B2(I,J+1,K+1)*VSEM2(4,I,J+1)/VSEM2(1,I,J+1)
519 ..... 1*-B2(I,J-1,K+1)*VSEM2(4,I,J-1)/VSEM2(1,I,J-1))/(2.*BY))
520 ..... 1-BATA*VSEM1(2,I,J)/VSEM1(1,I,J)*

```



```

521.....2*GF(I+1,J)-GF(I-1,J))
522.C
523.C.....ADD·NUMERICAL·DISSIPATION·TERM
524.....HON=0.10
525.....IF(VSEM1(1,I,J):LE:0.3)*HON=0.5
526.....GD(I,J)=GD(I,J)
527.....6+0.5/4:*(HON*BT/BX*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)
528.....7/VSEM1(1,I,J))*(GF(I+1,J)-GF(I,J))-ABS(VSEM1(2,I,J)/VSEM1(1,I,J)
529.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))*(GF(I,J)-GF(I-1,J)))
530.....6+0.5/4:*(HON*BT/BY*(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
531.....7/VSEM1(1,I,J))*(GF(I,J+1)-GF(I,J))-ABS(VSEM1(3,I,J)/VSEM1(1,I,J)
532.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))*(GF(I,J)-GF(I,J-1)))
533.....15·CONTINUE
534.....16·CONTINUE
535.....CALL·QACOPY(GF,XI(1,1,2),NS)
536.....CALL·CROUT(M)
537.....GO·TO·317
538.....312·CONTINUE
539.....DO·22·J=JB,JE
540.....DO·21·I=IB,IE
541.....GE(I,J)=ODE*SR(I)/(BX*BX*(GE(I-1,J)+
542.....1GE(I,J))/2:*(SR(I-1)+SR(I))/2:))
543.....5+BATA*VSEM2(2,I,J)/VSEM2(1,I,J)
544.....GA(I,J)=ODE*SR(I)/(BX*BX*(GE(I+1,J)+
545.....1GE(I,J))/2:*(SR(I+1)+SR(I))/2:))
546.....4-BATA*VSEM2(2,I,J)/VSEM2(1,I,J)
547.....GB(I,J)=1: +ODE*SR(I)/(BX*BX)*(2:/(GE(I+1,J)
548.....1+GE(I,J))*2:/(SR(I+1)+SR(I))+
549.....22:/(GE(I-1,J)+GE(I,J))*2:/(SR(I-1)+SR(I)))
550.....3+BT/2:0*(SR(I)/(2:0*BX)*
551.....4(VSEM2(2,I+1,J)/VSEM2(1,I+1,J)/SR(I+1)
552.....5-VSEM2(2,I-1,J)/VSEM2(1,I-1,J)/SR(I-1))
553.....6+(VSEM2(3,I,J+1)/VSEM2(1,I,J+1)-VSEM2(3,I,J-1)/VSEM2(1,I,J-1))
554.....1/(2:0*BY))
555.....GD(I,J)=GF(I,J)+ODE/(BY*BY)*((GF(I,J+1)-GF(I,J))/
556.....1*(T(I,J+1,K)+T(I,J,K))/2:))
557.....2-(GF(I,J)-GF(I,J-1))/((T(I,J-1,K)+T(I,J,K))/2:))
558.....3-BT/2:0*GF(I,J)*(SR(I)/(2:0*BX)*
559.....4(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)/SR(I+1)
560.....5-VSEM1(2,I-1,J)/VSEM1(1,I-1,J)/SR(I-1))
561.....6+(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))

```

```

562.....1/(2.0*BY)
563.....7+BT*SR(I)*((B1(I+1,J,K+1)+VSEM2(4,I+1,J)/VSEM2(1,I+1,J)
564.....8-B1(I-1,J,K+1)+VSEM2(4,I-1,J)/VSEM2(1,I-1,J))/(2.0*BK)
565.....9+(B2(I,J+1,K)+VSEM1(4,I,J+1)/VSEM1(1,I,J+1)
566.....1-B2(I,J-1,K)+VSEM1(4,I,J-1)/VSEM1(1,I,J-1))/(2.0*BY)
567.....2-BATB*VSEM1(3,I,J)/VSEM1(1,I,J)*
568.....3GF(I,J+1)-GF(I,J-1))
569..C
570..C.....ADD NUMERICAL DISSIPATION TERM
571.....HON=0.10
572.....IF(VSEM1(1,I,J).LE.0.3) HON=0.5
573.....GB(I,J)=GB(I,J)
574.....6+0.5/4.0*HON*BT/BK*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)
575.....7/VSEM1(1,I,J))*((GF(I+1,J)-GF(I,J))-ABS(VSEM1(2,I,J)/VSEM1(1,I,J)
576.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))*((GF(I,J)-GF(I-1,J)))
577.....6+0.5/4.0*HON*BT/BY*(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
578.....7/VSEM1(1,I,J))*((GF(I,J+1)-GF(I,J))-ABS(VSEM1(3,I,J)/VSEM1(1,I,J)
579.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))*((GF(I,J)-GF(I,J-1)))
580....21 CONTINUE
581....22 CONTINUE
582.....CALL QACOPY(GF,XI(1,1,2),NS)
583.....CALL CROUT(M)
584....317 CONTINUE
585.....DO 325 I=IS,IB
586.....GF(I,JB-1)=GF(I,JB+1)
587....325 CONTINUE
588.....DO 332 I=IB,IE
589.....DO 331 J=JB,JE
590.....B3(I,J,K+1)=GF(I,J)/SR(I)
591.....C1(I,J,K+1)=- (GF(I,J+1)-GF(I,J-1))/(SR(I)+2.0*BY)
592.....C2(I,J,K+1)= (GF(I+1,J)-GF(I-1,J))/(SR(I)+2.0*BK)
593....331 CONTINUE
594.....C1(I,1,K+1)=-C1(I,3,K+1)
595.....C2(I,1,K+1)=+C2(I,3,K+1)
596....332 CONTINUE
597.....DO 133 I=IS,IB
598.....B3(I,JB,K+1)=GF(I,JB)/SR(I)
599.....B3(I,JS,K+1)=GF(I,JS)/SR(I)
600....133 CONTINUE

```

```

601 ..... DD 134 J=JS, JB
602 ..... B3 (IB, J, K+1) = GF (IB, J) / SR (IB)
603 ..... B3 (IS, J, K+1) = GF (IS, J) / SR (IS)
604 ..... 134 CONTINUE
605 ..... CALL QACOPY (XI (1, 1, 2), GF, NS)
606 ..... CALL QACOPY (GE, CN (1, 1, 2), NS)
607 ..... CALL QACOPY (GF, EI (1, 1, 1), NS)
608 ..... IF (MOD (M, 2), NE, 1) GO TO 212
609 ..... 211 CONTINUE
610 ..... DD 214 J=JB, JE
611 ..... DD 213 I=IB, IE
612 ..... GC (I, J) = BT * (GE (I, J) +
613 ..... 1GE (I, J-1)) / (2. * VSEM2 (1, I, J) +
614 ..... 2BY * BY) + BT * VSEM2 (3, I, J) /
615 ..... 3VSEM2 (1, I, J) / (2. * BY)
616 ..... GA (I, J) = BT * (GE (I, J) +
617 ..... 1GE (I, J+1)) / (2. * VSEM2 (1, I, J) +
618 ..... 2BY * BY) - BT * VSEM2 (3, I, J) /
619 ..... 3VSEM2 (1, I, J) / (2. * BY)
620 ..... GB (I, J) = 1. + BT / (VSEM2 (1, I, J) * BY * BY) * (
621 ..... 1 * (GE (I, J) + GE (I, J+1)) / 2.
622 ..... 2 * (GE (I, J) + GE (I, J-1)) / 2.)
623 ..... 4 + BT * (FK-1.) * 1. * ((SR (I+1) * VSEM2 (2, I+1, J) / VSEM2 (1, I+1, J)
624 ..... 5 - SR (I-1) * VSEM2 (2, I-1, J) / VSEM2 (1, I-1, J)) / (2. * SR (I) * BX)
625 ..... 7 + (VSEM2 (3, I, J+1) / VSEM2 (1, I, J+1) -
626 ..... 8VSEM2 (3, I, J-1) / VSEM2 (1, I, J-1)) / (2. * BY))
627 ..... GB (I, J) = GF (I, J) + BT / (VSEM1 (1, I, J) * SR (I) * BX * BX) * (0.5 * (SR (I+1) + SR (I))
628 ..... 2 * 0.5 * (CN (I, J, K) + CN (I+1, J, K)) * (GF (I+1, J) - GF (I, J)) - 0.5 * (SR (I-1)
629 ..... 4 + SR (I)) * 0.5 * (CN (I, J, K) + CN (I-1, J, K)) * (GF (I, J) - GF (I-1, J)))
630 ..... 5 - BT * VSEM1 (2, I, J) / VSEM1 (1, I, J)
631 ..... 6 * (GF (I+1, J) - GF (I-1, J)) / (2. * BX)
632 ..... 8 + CDE * (E1 (I, J, K+1) * 2 + E2 (I, J, K+1) * 2 + E3 (I, J, K+1) * 2) /
633 ..... 9 (T (I, J, K) * VSEM2 (1, I, J))
634 ..... 9 - 0.8 * BT * (FK-1.) * GF (I, J) *
635 ..... 4 * (SR (I+1) * VSEM1 (2, I+1, J) / VSEM1 (1, I+1, J)
636 ..... 5 - SR (I-1) * VSEM1 (2, I-1, J) / VSEM1 (1, I-1, J)
637 ..... 6) / (2. * SR (I) * BX)
638 ..... 7 + (VSEM1 (3, I, J+1) / VSEM1 (1, I, J+1) -
639 ..... 8VSEM1 (3, I, J-1) / VSEM1 (1, I, J-1)) / (
640 ..... 92. * BY))

```

```

641
642 C.....ADD NUMERICAL DISSIPATION TERM
643.....HON=0.10
644.....IF (VSEM1(1,I,J).LE.0.3) HON=0.5
645.....GB(I,J)=GB(I,J)
646.....6+0.5/4.*HON*BT/BX*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)
647.....7/VSEM1(1,I,J))*GF(I+1,J)-GF(I,J))-ABS(VSEM1(2,I,J)/VSEM1(1,I,J)
648.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))*GF(I,J)-GF(I-1,J))
649.....6+0.5/4.*HON*BT/BY*(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
650.....7/VSEM1(1,I,J))*GF(I,J+1)-GF(I,J))-ABS(VSEM1(3,I,J)/VSEM1(1,I,J)
651.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))*GF(I,J)-GF(I,J-1))
652...213 CONTINUE
653...214 CONTINUE
654.....CALL QACOPY(GF,EI(1,1,2),NS)
655.....CALL CROUT(M)
656.....CALL QACOPY(EI(1,1,2),GF,NS)
657.....GO TO 217
658...212 CONTINUE
659.....DO 235 J=JB,JE
660.....DO 234 I=IB,IE
661.....GC(I,J)=BT*0.5*(GE(I,J)+GE(I-1,J))*0.5*(SR(I-1)+SR(I))/
662.....2(VSEM2(1,I,J)*SR(I)*BX*BX)+BT*VSEM2(2,I,J)/
663.....3VSEM2(1,I,J)/(2.*BX)
664.....GA(I,J)=BT*0.5*(GE(I,J)+GE(I+1,J))*0.5*(SR(I+1)+SR(I))/
665.....2(VSEM2(1,I,J)*SR(I)*BX*BX)-BT*VSEM2(2,I,J)/
666.....3VSEM2(1,I,J)/(2.*BX)
667.....GB(I,J)=1.+BT/(VSEM2(1,I,J)*SR(I)*BX*BX)*(
668.....10.5*(GE(I,J)+GE(I+1,J))*0.5*(SR(I)+SR(I+1))+
669.....30.5*(GE(I,J)+GE(I-1,J))*0.5*(SR(I)+SR(I-1)))
670.....4+BT*(FK-1.)*1.0*((SR(I+1)*VSEM2(2,I+1,J)/
671.....5VSEM2(1,I+1,J)-SR(I-1)*W(2,I-1,J)
672.....6K+1)/VSEM2(1,I-1,J))/(2.*SR(I)*BX)+
673.....7(VSEM2(3,I,J+1)/VSEM2(1,I,J+1)-
674.....8VSEM2(3,I,J-1)/VSEM2(1,I,J-1))/
675.....9(2.*BY))
676.....GD(I,J)=GF(I,J)+BT/(VSEM1(1,I,J)*BY*BY)*(
677.....10.5*(EN(I,J,K)+EN(I,J+1,K))*GF(I,J+1)-
678.....2GF(I,J))-0.5*(EN(I,J,K)+
679.....3EN(I,J-1,K))*GF(I,J)-GF(I,J-1))
680.....4-BT*VSEM1(3,I,J)/VSEM1(1,I,J)*GF(I,J+1)-

```

```

681 ..... 5GF(I,J-1))/(2.*BY)
682 ..... 8+ODE*(C1(I,J,K+1)**2+C2(I,J,K+1)**2+C3(I,J,K+1)**2)/
683 ..... 9(T(I,J,K-1)*VSCM2(1,I,J))
684 ..... 9.....-BT*(FK-1.)*GF(I,J)*0.0*
685 ..... 4((SR(I+1)*VSCM1(2,I+1,J)/VSCM1(1,I+1,J)
686 ..... 5.-SR(I-1)*VSCM1(2,I-1,J)/VSCM1(1,I-1,J)
687 ..... 6)/(2.*SR(I)*BX))+
688 ..... 7VSCM1(3,I,J+1)/VSCM1(1,I,J+1)-
689 ..... 8VSCM1(3,I,J-1)/VSCM1(1,I,J-1))/(
690 ..... 92.*BY))
691 C
692 C..... ADD NUMERICAL DISSIPATION TERM
693 ..... HDN=0.10
694 ..... IF(VSCM1(1,I,J).LE.0.3) HDN=0.5
695 ..... GB(I,J)=GB(I,J)
696 ..... 6+0.5/4.*HDN*BT/BX*(ABS(VSCM1(2,I+1,J)/VSCM1(1,I+1,J)-VSCM1(2,I,J)
697 ..... 7/VSCM1(1,I,J))*(GF(I+1,J)-GF(I,J))-ABS(VSCM1(2,I,J)/VSCM1(1,I,J)
698 ..... 8-VSCM1(2,I-1,J)/VSCM1(1,I-1,J))*(GF(I,J)-GF(I-1,J)))
699 ..... 6+0.5/4.*HDN*BT/BY*(ABS(VSCM1(3,I,J+1)/VSCM1(1,I,J+1)-VSCM1(3,I,J)
700 ..... 7/VSCM1(1,I,J))*(GF(I,J+1)-GF(I,J))-ABS(VSCM1(3,I,J)/VSCM1(1,I,J)
701 ..... 8-VSCM1(3,I,J-1)/VSCM1(1,I,J-1))*(GF(I,J)-GF(I,J-1)))
702 ...234 CONTINUE
703 ...235 CONTINUE
704 ..... CALL QACOPY(GF,EI(1,1,2),NS)
705 ..... CALL GROUT(M)
706 ..... CALL QACOPY(EI(1,1,2),GF,NS)
707 ...217 CONTINUE
708 ..... DO 225 I=IS,IB
709 ..... EI(I,JB-1,K+1)=EI(I,JB+1,K+1)
710 ...225 CONTINUE
711 ..... RETURN
712 ..... END
713
714
715
716
717
718
719 ..... SUBROUTINE GROUT(M)
720 C..... THIS SUBROUTINE IS USED BY "ADI" TO INVERT

```

```

721 C.....A TRI-DIAGONAL MATRIX USING CROUT'S METHOD
722 .....LCM (VARIABLES), (LCMICCG), (COEFAD), (CONTOUR), (WLCF2)
723 .....COMMON /WSEM/ S(N, (-4,4))
724 .....COMMON /WLCM2/ GA(IN2, JM1P1), GB(IN2, JM1P1), GC(IN2, JM1P1),
725 .....1GD(IN2, JM1P1), GE(IN2, JM1P1), GF(IN2, JM1P1), VSEM1(4, IN2, JM1P1),
726 .....2VSEM2(4, IN2, JM1P1), APSI(NT, NT)
727 .....COMMON /CONST2/ IS, IB, IE, ID, IP, JS, JB, JE, JD, JP, DX, EY, BATA, BATB
728 .....DIMENSION AA(IN2, 3), E(IN2), X(IN2)
729 .....OPTIMIZE
730 .....SMALL=1.0E-30
731 .....IF (MOD(M,2).NE.1) GO TO 2
732 .....JQ=JP-1
733 .....DO 123 I=IB, IE
734 .....DO 122 J=JB, JQ
735 .....AA(J,1)=-GC(I, J+1)
736 .....AA(J,2)=+GB(I, J+1)
737 .....AA(J,3)=-GA(I, J+1)
738 .....122 E(J) =+GB(I, J+1)
739 .....E(JS) =GB(I, JB)
740 .....AA(JS,2)=GB(I, JB)
741 .....AA(JS,3)=- (GA(I, JB)+GC(I, JB))
742 .....E(JP) =+GB(I, JE)+GA(I, JE)+GF(I, JE+1)
743 .....AA(JP,2)=GB(I, JE)
744 .....AA(JP,1)=-GC(I, JE)
745 .....AA(JS,3...) =AA(JS,3...)/AA(JS,2)
746 .....E(JS) =E(JS)/AA(JS,2)
747 .....DO 130 J=JB, JP
748 .....IF (ABS(AA(J,...1)) .LE. SMALL) AA(J,...1)=0.
749 .....IF (ABS(AA(J-1,3)) .LE. SMALL) AA(J-1,3)=0.
750 .....AA(J,2)=AA(J,2)-AA(J,...1)*AA(J-1,3)
751 .....IF (ABS(E(J-1)) .LE. SMALL) E(J-1)=0.
752 .....E(J) = (E(J)-AA(J,...1)*E(J-1))/AA(J,2)
753 .....IF (J.EQ. JP) GO TO 131
754 .....AA(J,3...) =AA(J,3...)/AA(J,2)
755 .....131 CONTINUE
756 .....130 CONTINUE
757 .....X(JP)=E(JP)
758 .....DO 133 J=JS, JQ
759 .....KK=JQ+JS-J
760 .....IF (ABS(AA(KK,3...)) .LE. SMALL) AA(KK,3...)=0.

```

```

761.....IF (ABS(X(KK+1)) .LE. SMALL) X(KK+1)=0.
762.....X(KK)=E(KK)-AA(KK,3...)*X(KK+1)
763....133 CONTINUE
764.....DO 128 J=JS,JP
765.....GF(I,J+1)=X(J)
766.....IF (ABS(GF(I,J+1)) .LT. SMALL) GF(I,J+1)=0.
767....128 CONTINUE
768....123 CONTINUE
769.....GO TO 4
770....2 CONTINUE
771.....IQ=IP-1
772.....DO 23 J=JB,JE
773.....DO 22 I=IB,IQ
774.....E(I) = GB(I+1,J)
775.....AA(I,1)=-GC(I+1,J)
776.....AA(I,2) = GB(I+1,J)
777.....AA(I,3)=-GA(I+1,J)
778....22 CONTINUE
779.....E(IS)=GB(IB,J)+GC(IB,J)+GF(IS,J)
780.....AA(IS,2)=GB(IB,J)
781.....AA(IS,3)=-GA(IB,J)
782.....E(IP)=GB(IE,J)+GA(IE,J)+GF(IE+1,J)
783.....AA(IP,1)=-GC(IE,J)
784.....AA(IP,2) = GB(IE,J)
785.....AA(IS,3...) = AA(IS,3...) / AA(IS,2)
786.....E(IS)=E(IS) / AA(IS,2)
787.....DO 30 I=IB,IP
788.....IF (ABS(AA(I,...1)) .LE. SMALL) AA(I,...1)=0.
789.....IF (ABS(AA(I-1,3)) .LE. SMALL) AA(I-1,3)=0.
790.....AA(I,2)=AA(I,2)-AA(I,...1)*AA(I-1,3)
791.....IF (ABS(E(I-1)) .LE. SMALL) E(I-1)=0.
792.....E(I)=(E(I)-AA(I,...1)*E(I-1)) / AA(I,2)
793.....IF (I.EQ. IP) GO TO 31
794.....AA(I,3...) = AA(I,3...) / AA(I,2)
795....31 CONTINUE
796....30 CONTINUE
797.....X(IP)=E(IP)
798.....DO 33 I=IS,IQ
799.....KK=IQ+IS-I
800.....IF (ABS(AA(KK,3...)) .LE. SMALL) AA(KK,3...) = 0.

```

```

801.....IF (ABS(X(KK+1)) .LE. SMALL) X(KK+1)=0.
802.....X(KK)=E(KK)-AA(KK,3) *X(KK+1)
803.....33-CONTINUE
804.....DO 28 I=IS,IP
805.....GF(I+1,J)=X(I)
806.....IF (ABS(GF(I+1,J)) .LT. SMALL) GF(I+1,J)=0.
807.....28-CONTINUE
808.....23-CONTINUE
809.....4-CONTINUE
810.....RETURN
811.....END
812
813
814
815
816
817
818.....SUBROUTINE CGCDEF
819 C..... THIS SUBROUTINE CALCULATES THE COEFFICIENT MATRIX
820 C..... USED BY "ICLU" TO CALCULATE PSI IN VACUUM
821..... LCM (VARIABLES), (LCMICCG), (COEFAD), (CONTOUR), (WLEN2)
822..... COMMON/VARIABLES/PSI(IN2,JM1P1,3), B1(IN2,JM1P1,2),
823..... 1B2(IN2,JM1P1,2), B3(IN2,JM1P1,2), X1(IN2,JM1P1,2), C1(IN2,JM1P1,2),
824..... 1C2(IN2,JM1P1,2), C3(IN2,JM1P1,2),
825..... 2T(IN2,JM1P1,2), W(4,IN2,JM1P1,3), F(4,IN2,JM1P1,3),
826..... 3EI(IN2,JM1P1,2), CH(IN2,JM1P1,2)
827..... COMMON/CONST2/IS,IB,IE,ID,IP,JS,JB,JE,JD,JP,BX,BY,BATX,BATY
828..... COMMON /MESH/ RR(NT),ZZ(MT),DELRR(NT-1),DELZZ(MT-1),
829..... 1SR(IN2),SZ(JM1P1),DELR(IN2-1),DELZ(JM1P1-1)
830..... COMMON/CROUTV/ AA(IN2,3), B(IN2), X1(IN2)
831..... COMMON/CONST/ N1,N2,N3,M1,M2,N1P1,N1P2,N2P1,N2P2,N2M1,
832..... 1N1PN2,N1PN2P,N1PN2M,M1P1,M1M1,M1P2,M2P1,M2M1,M2P2,NH,NHM1,MM,MMM1
833..... COMMON/CUR/ NCUR,LOCUR(35),VALCUR(35)
834..... COMMON/DIM/RMA,RB,BB1,RLNGTH,ZLNGTH
835..... COMMON/COEFAD/COEFR1(IN2-1,JM1P1-1),COEFR2(IN2-1,JM1P1-1),
836..... 2COEFR3(IN2-1,JM1P1-1),COEFZ1(IN2-1,JM1P1-1),
837..... 3COEFZ2(IN2-1,JM1P1-1),COEFZ3(IN2-1,JM1P1-1)
838..... COMMON/CONTOUR/ FLUXR2(NT,MT),RAXIS(NT),ZAXIS(MT)
839..... COMMON/VAR2/FR,FK,CY,C,Q,HU,UN,BT,TMAX,PTIME1,PTIME2,TIME,
840..... 1 TBIS1,TBIS2,

```



```

841 ..... 1BMIN,SGO,CDE,MW,MP,MC,NCURV,NCHECK,BTOR,BN,TEMP,
842 ..... 2CUR1MAX,CUR2MAX,CUR3MAX,CURTOR,RSPLNGH,RBPNGH
843 ..... COMMON /WSEM/ S(N,(-4,4))
844 ..... COMMON /WLCM2/ GA(IN2,JM1P1),GB(IN2,JM1P1),GC(IN2,JM1P1),
845 ..... 1GD(IN2,JM1P1),GE(IN2,JM1P1),GF(IN2,JM1P1),VSEM1(4,IN2,JM1P1),
846 ..... 2VSEM2(4,IN2,JM1P1),APSI(MT,MT)
847 ..... COMMON /LCMICG6/ MTRX(N,(-4,4)),LU(N,(-4,4)),X(N),Y(N)
848 ..... REAL MTRX,LU
849 C.....
850 ..... OPTIMIZE
851 ..... KK(I,J)=(I-2)*(MT-2)+(J-1)
852 ..... CALL QYSET(0.,S,9*N)
853 ..... DO LOOP1 I=2,N1
854 ..... DO LOOP1 J=2,MMM1
855 ..... K=KK(I,J) $ CALL CDEF(I,J,XA,XB,XC,XD,XE)
856 ..... IF(I.NE.2) S(K,-3)=XA
857 ..... IF(J.NE.2) S(K,-1)=XB
858 ..... S(K,0)=XC $ S(K,+1)=XE
859 LOOP1 S(K,+3)=XD
860 C..... TAKE CARE OF THE BOUNDARY AT I=N1P1
861 ..... I=N1P1 $ DO LOOP2 J=2,M2
862 ..... K=KK(I,J) $ CALL CDEF(I,J,XA,XB,XC,XD,XE)
863 ..... S(K,-3)=XA $ IF(J.NE.2) S(K,-1)=XB
864 ..... S(K,0)=XC
865 ..... S(K,+1)=XE
866 LOOP2 S(K,+3)=XD
867 ..... DO LOOP4 I=N1P1,NNM1
868 ..... DO LOOP5 J=2,M2
869 ..... K=KK(I,J) $ CALL CDEF(I,J,XA,XB,XC,XD,XE)
870 ..... S(K,-3)=XA $ IF(J.NE.2) S(K,-1)=XB
871 ..... S(K,0)=XC $ S(K,+1)=XE
872 ..... IF(I.NE.NNM1) S(K,+3)=XD
873 LOOP5 CONTINUE
874 ..... DO LOOP6 J=M2P1,MMM1
875 ..... K=KK(I,J)
876 LOOP6 S(K,0)=1.0
877 LOOP4 CONTINUE
878 C..... TAKE CARE OF THE INNER BOUNDARY...
879 ..... I=N1 $ DO 10 J=M2P1,MMM1
880 ..... K=KK(I,J)

```

```

881 .....10 S(K,+3)=0.
882 .....J=M2...$...DO 11 I=N1P1,MM1
883 .....K=KK(I,J)
884 .....11 S(K,+1)=0.
885 C.....END OF COEFFICIENT CALCULATIONS
886 .....CALL QACOPY(MTRX,S,9*M)
887 .....CALL QVSET(0.,S,9*M)
888 .....RETURN
889 .....END
890
891
892
893
894
895
896 .....SUBROUTINE CONTR(IMIN,IMAX,JMIN,JMAX,NCURV,ISW,C,M)
897 C.....THIS SUBROUTINE IS CALLED BY *CLPLOT* TO PLOT
898 C.....CONTOURS OF DENSITY, FLUX FUNCTION PSI, ETC.
899 C.....LCM (VARIABLES), (LCMIGG), (COEFAD), (CONTOUR), (JLCM2)
900 C.....COMMON /CONTOUR/ FLUXRZ(MT,MT),RAXIS(MT),ZAXIS(MT)
901 C.....DIMENSION C(35),CPR(35)
902 C.....COMMON /CONST/ N1,N2,N3,M1,M2,N1P1,N1P2,N2P1,N2P2,N2M1,
903 C.....1N1PN2,N1PN2P,N1PN2M,M1P1,M1M1,M1P2,M2P1,M2M1,M2P2,NN,MM1,MM,MMM1
904 C.....COMMON /DIM/ RSMA,RB,BB1,RLNGTH,ZLNGTH
905 C.....NCURV:* OF CONTOURS TO BE DRAWN.
906 C.....CONTOURS ARE DRAWN IN THE DOMAIN SPECIFIED BY IMIN,IMAX
907 C.....JMIN,JMAX.
908 C.....FIND THE MIN. AND THE MAX. VALUES OF FLUX IN THE DOMAIN
909 C.....OF INTEREST.
910 .....IF (ISW.GT.0) GO TO 20...$$$ TRY TO PLOT THE CLOSED FIELD LINES.
911 .....PSIMAX=-1.0E+200
912 .....PSIMIN=+1.0E+200
913 .....DO 2 I=IMIN,IMAX
914 .....DO 2 J=JMIN,JMAX
915 .....IF (FLUXRZ(I,J).GT.PSIMAX) PSIMAX=FLUXRZ(I,J)
916 .....IF (FLUXRZ(I,J).LT.PSIMIN) PSIMIN=FLUXRZ(I,J)
917 .....2 CONTINUE
918 .....DEL=(PSIMAX-PSIMIN)/(NCURV-1)
919 .....DO 4 L=1,NCURV
920 .....4 C(L)=PSIMIN+(L-1)*DEL

```

```

921.....GO TO 40
922....20 IF (M.GT.1) GO TO 25
923.....DO 21 I=1,N2-1
924....21 C(I)=FLUXRZ(I,2)
925.....PSIMAX=AMAXAF(C,1,N2-1) PSIMIN=AMINAF(C,1,N2-1)
926.....DEL=(PSIMAX-PSIMIN)/(NCURY-1)
927.....DO 26 L=1,NCURY
928.....C(L)=PSIMIN+(L-1)*DEL
929....26 CPER(L)=C(L)
930.....GO TO 40
931....25 DO 27 L=1,NCURY
932....27 C(L)=CPER(L)
933....40 NUM=NCURY
934 C.....NOW WE ARE READY TO PLOT:
935.....XMIN=RAXIS(IMIN)
936.....XMAX=RAXIS(IMAX)
937.....YMIN=ZAXIS(JMIN)
938.....YMAX=ZAXIS(JMAX)
939.....CALL FRAME
940.....RATIO=(YMAX-YMIN)/(XMAX-XMIN)
941.....IF (RATIO.GT.0.9) GO TO 10
942.....CALL MAP(XMIN,XMAX,YMIN,YMAX,0.,0.8,0.1,0.1+0.8*RATIO)
943.....GO TO 12
944....10 RATIO=1./RATIO
945.....CALL MAP(XMIN,XMAX,YMIN,YMAX,0.,0.8*RATIO,0.1,0.9)
946 C.....DRAW THE POSITION OF THE INNER BOUNDARY
947....12 XMINP=RAXIS(1)
948.....XMAXP=RAXIS(N2)-RAXIS(N2)/1000.
949.....YMINP=ZAXIS(2)
950.....YMAXP=ZAXIS(M1+1)
951.....CALL LINE(XMINP,YMINP,XMAXP,YMINP)
952.....CALL LINE(XMAXP,YMINP,XMAXP,YMAXP)
953.....CALL LINE(XMAXP,YMAXP,XMINP,YMAXP)
954.....CALL LINE(XMINP,YMAXP,XMINP,YMINP)
955 C.....DRAW THE CONTOURS NOW:
956.....CALL RCONTR(NUM,C,NUM+1,FLUXRZ,NT,RAXIS,IMIN,
957....1IMAX,1,ZAXIS,JMIN,JMAX,1)
958.....RETURN
959.....END
960

```

```

966.....SUBROUTINE FLX(M,LERR)
967 C..... THIS SUBROUTINE CALCULATES THE FLUX FUNCTION PSI
968 C..... IN THE PLASMA AND THE VACUUM REGIONS
969..... LCM (VARIABLES), (LCMICCG), (CDEFAB), (CONTOUR), (WLCM2)
970..... COMMON/VARIABLES/PSI (IN2,JM1P1,3), B1 (IN2,JM1P1,2),
971..... 1B2 (IN2,JM1P1,2), B3 (IN2,JM1P1,2), X1 (IN2,JM1P1,2), C1 (IN2,JM1P1,2),
972..... 1C2 (IN2,JM1P1,2), C3 (IN2,JM1P1,2),
973..... 2T (IN2,JM1P1,2), W(4,IN2,JM1P1,3), F(4,IN2,JM1P1,3),
974..... 3EI (IN2,JM1P1,2), CN (IN2,JM1P1,2)
975..... COMMON/CONST2/IS, IB, IE, IB, IP, JS, JB, JE, JB, JP, BX, BY, BATA, BATB
976..... COMMON /MESH/ RR (NT), ZZ (MT), DELRR (NT-1), DELZZ (MT-1),
977..... 1SR (IN2), SZ (JM1P1), DELR (IN2-1), DELZ (JM1P1-1)
978..... COMMON/CROUTV/ AA (IN2,3), D (IN2), X1 (IN2)
979..... COMMON/CONST/ N1,N2,N3,M1,M2,N1P1,N1P2,N2P1,N2P2,N2M1,
980..... 1N1PN2,N1PN2P,N1PN2M,M1P1,M1M1,M1P2,M2P1,M2M1,M2P2,NN,NNM1,MM,MMM1
981..... COMMON/CUR/ NCUR,LOCUR(35),VALCUR(35)
982..... COMMON/BIM/RSMA,RB,BB1,RLNGTH,ZLNGTH
983..... COMMON/CDEFAB/CDEFR1 (IN2-1,JM1P1-1),CDEFR2 (IN2-1,JM1P1-1),
984..... 2CDEFR3 (IN2-1,JM1P1-1),CDEFZ1 (IN2-1,JM1P1-1),
985..... 3CDEFZ2 (IN2-1,JM1P1-1),CDEFZ3 (IN2-1,JM1P1-1)
986..... COMMON/CONTOUR/ FLUXR2 (NT,MT),RAXIS (NT),ZAXIS (MT)
987..... COMMON/VAR2/FR,FK,CV,C,Q,HU,UN,BT,TMAX,PTIME1,PTIME2,TIME,
988..... 1 TDIS1,TDIS2,
989..... 1BMIN,SGO,CDE,MW,MP,MC,NCURY,NCHECK,BTOR,BN,TEMP,
990..... 2CUR1MAX,CUR2MAX,CUR3MAX,CURTOR,RSPINCH,RBPINCH
991..... COMMON /WSCM/ S(N,(-4,4))
992..... COMMON/WLCM2/ GA (IN2,JM1P1),GB (IN2,JM1P1),GC (IN2,JM1P1),
993..... 1GB (IN2,JM1P1),GE (IN2,JM1P1),GF (IN2,JM1P1),VSCM1(4,IN2,JM1P1),
994..... 2VSCM2(4,IN2,JM1P1),APSI (NT,MT)
995..... COMMON /LCMICCG/ MTRX(N,(-4,4)),LU(N,(-4,4)),X(N),Y(N)
996..... REAL MTRX,LU
997 C..... *****
998 C..... CALCULATE NEW COEFFICIENT MATRICES FOR ABIPSI ROUTINE
999..... OPTIMIZE
1000..... KK(I,J)=(I-2)*(MT-2)+(J-1)
1001..... LERR=0
1002..... SMALL=1.0E-30
1003..... DD=500 J=2,M1
1004..... JJ=J-1
1005..... DD=501 I=2,N2M1

```

```

1006.....II=I-1
1007.....CDEFR1(II,JJ)=CDE*SR(I)
1008.....1/(T(I,J,2)*BX*BX*(SR(I-1)+SR(I))/2.)
1009.....2+BT*W(2,I,J,2)/W(1,I,J,2)/(2.*BX)
1010.....CDEFR3(II,JJ)=CDE*SR(I)
1011.....1/(T(I,J,2)*BX*BX*(SR(I+1)+SR(I))/2.)
1012.....2-BT*W(2,I,J,2)/W(1,I,J,2)/(2.*BX)
1013.....CDEFR2(II,JJ)=1.+CDE/(T(I,J,2)*BX*BX)
1014.....1*SR(I)+(2./(SR(I+1)+SR(I))+2./(SR(I-1)+SR(I)))
1015...501 CONTINUE
1016...500 CONTINUE
1017...502 DO 510 I=2,N2M1
1018.....II=I-1
1019.....DO 512 J=2,M1
1020.....JJ=J-1
1021.....CDEFZ1(II,JJ)=CDE/(T(I,J,2)*BY*BY)
1022.....1+BT*W(3,I,J,2)/W(1,I,J,2)/(2.*BY)
1023.....CDEFZ3(II,JJ)=CDE/(T(I,J,2)*BY*BY)
1024.....1-BT*W(3,I,J,2)/W(1,I,J,2)/(2.*BY)
1025.....CDEFZ2(II,JJ)=1.+2.*CDE/(T(I,J,2)*BY*BY)
1026...512 CONTINUE
1027...510 CONTINUE
1028 C
1029 C.....END OF COEFFICIENT CALCULATIONS.
1030...503 ITERAD=0
1031.....ITERCG=0
1032.....EPSCG=1.0E-3
1033.....IF (M.LT.10) EPSCG=1.0E-2
1034.....EPSABI=1.0E-4
1035.....IF (M.LT.50) EPSABI=1.0E-3
1036.....IF (M.LT.10) EPSABI=1.0E-2
1037 C.....START OF THE INNER ITERATIONS...
1038...701 KFLAG=0
1039 C.... GET NEW X(K) ....
1040.....DO 410 I=2,N2M1
1041.....DO 410 J=2,M2M1
1042.....K=KK(I,J)
1043...410 S(K,0)=APSI(I,J)
1044.....CALL QACOPY(X,S(1,0),M)
1045 C.... CALCULATE Y(K) ....

```

```

1046 .....CALL QVSET(0.,S(1,0),N)
1047 .....DO 403 L=1,NCUR
1048 .....IJ=LOCUR(L)
1049 .....I=IJ/100.$..J=IJ-100*I
1050 .....K=KK(I,J)
1051 ...403 S(K,0)=(1.0-RLNGTH*RR(I))/RR(I)*VALCUR(L)
1052 .....I=N1.$..DO 401 J=M2P1,MMM1
1053 .....CALL QDEF(I,J,XA,XB,XC,XD,XE)
1054 .....K=KK(I,J)
1055 ...401 S(K,0)=S(K,0)-XB*APSI(I+1,J)
1056 .....J=M2.$..DO 402 I=N1P1,NNM1
1057 .....CALL QDEF(I,J,XA,XB,XC,XD,XE)
1058 .....K=KK(I,J)
1059 ...402 S(K,0)=S(K,0)-XE*APSI(I,J+1)
1060 .....DO 404 I=N1P1,NNM1
1061 .....DO 404 J=M2P1,MMM1
1062 .....K=KK(I,J)
1063 ...404 S(K,0)=APSI(I,J)
1064 .....CALL QACOPY(Y,S(1,0),N)
1065 C....END OF Y(K) CALCULATIONS....
1066 ...723 CALL QGITER(KFLAG,BXNORM,RNORM)
1067 .....ITERCG=ITERCG+1
1068 .....IF (BXNORM.GT.EPSCG) GO TO 723
1069 .....CALL QACOPY(S(1,0),X,N)
1070 .....DO 721 I=2,NNM1
1071 .....DO 722 J=2,MMM1
1072 .....K=KK(I,J)
1073 ...722 APSI(I,J)=S(K,0)
1074 ...721 APSI(I,MM)=APSI(I,MM-2)
1075 C....GET THE NEW BNDRY VAL. S FOR PSI...
1076 .....CALL BNDRY
1077 ...750 ITERAD=ITERAD+1
1078 .....IF (ITERAD.LT.3000) GO TO 753
1079 .....WRITE(3,754) M,ITERCG,ITERAD,BXNORM,ERRM
1080 ...754 FORMAT(///5X,"ITERAD EXCEEDED LIMIT."
1081 .....1/5X,"M=",15/5X,"ITERCG=",15/5X,"ITERAD=",15/5X,
1082 .....1"BXNORM=",E12.4/5X,"ERRM=",E12.4)
1083 .....LERR=1.$..RETURN
1084 ...753 CALL ADIPSI(M)
1085 .....ERRM=-1.0E+200

```

```

1086.....ERR=0.
1087.....DO 751 I=2,N2-2
1088.....DO 752 J=2,M1P1-2
1089.....IF (ABS (PSI (I,J,3)) .LT. SMALL) GO TO 752
1090.....ERR=ABS (1.0-PSI (I,J,2) / PSI (I,J,3))
1091.....IF (ERR.GT.ERRM) ERRM=ERR
1092...752 CONTINUE
1093...751 CONTINUE
1094.....TRANSFER....
1095.....CALL @ACOPY (GA,PSI (1,1,3),NS)
1096.....CALL @ACOPY (PSI (1,1,2),GA,NS)
1097.....COPY PSI TO APSI....
1098.....DO 730 I=1,N2
1099.....DO 730 J=1,M1P1
1100.....IW=NN-I+1,JW=MM-J+1
1101...730 APSI (IW,JW)=GA (I,J)
1102.....CALL BNDY...$$$...CALCULATE THE NEW BNDY VAL.
1103.....IF (M.LT.10.AND.ITERAD.LT.2) GO TO 758
1104.....XA=ERRM/2.
1105.....IF (XA.LT.EPSABI) XA=EPSABI
1106.....IF (XA.LT.EPSCG) EPSCG=XA
1107...758 IF (ERRM.GT.EPSABI.OR.ERRM.LT.0.) GO TO 701
1108.....IF (MOD (M,50).NE.1) GO TO 760
1109.....WRITE (3,777) M,ITERCG,ITERAD
1110...777 FORMAT (/5X,"M=",I5,5X,"ITERCG=",I5,5X,"ITERAD=",I5/)
1111.....INNER ITERATIONS HAVE CONVERGED.CALCULATE THE NEW
1112.....TOROIDAL CURRENT.
1113...760 DO 761 J=1,M2P1
1114.....DO 761 I=1,NN
1115...761 FLUXRZ (NN+1-I,MM+1-J)=APSI (I,J)
1116.....DO 762 J=M2P2,MM
1117.....DO 763 I=1,M1
1118...763 FLUXRZ (NN+1-I,MM+1-J)=APSI (I,J)
1119...762 CONTINUE
1120.....DO 765 I=1,N2
1121.....DO 765 J=1,M1P1
1122...765 FLUXRZ (I,J)=GA (I,J)
1123.....CALL @VSET (0.,GB,NS)
1124.....DO 27 J=2,M1
1125.....DO 26 I=2,N2M1

```

```

1126.....GB(I,J)=-(((GA(I+1,J)-GA(I,J))/((SR(I+1)+SR(I))/2.)
1127.....1-(GA(I,J)-GA(I-1,J))/((SR(I-1)+SR(I))/2.)
1128.....2)/(BX+BX)+((GA(I,J+1)-GA(I,J))
1129.....3-(GA(I,J)-GA(I,J-1)))/(SR(I)+BY+BY))
1130.....26-CONTINUE
1131.....27-CONTINUE
1132.....DO 32 J=2,M1P1
1133.....GC(I,J)=0;...$$$-BR-IS-ZERO-ON-THR-INNER-WALL;
1134.....DO 32 I=2,N2
1135.....GC(I,J)=-(FLUXRZ(I,J+1)-FLUXRZ(I,J-1))/(2.0+SR(I)+BY)
1136.....GB(I,J)=+(FLUXRZ(I+1,J)-FLUXRZ(I-1,J))/
1137.....1(SR(I)+(RAXIS(I+1)-RAXIS(I-1)))
1138.....32-CONTINUE
1139.....DO 35 J=2,M1P1
1140.....GB(N2,J)=(FLUXRZ(N2,J)-FLUXRZ(N2-1,J))/(SR(N2)+BX)
1141.....35-GB(1,J)=FLUXRZ(1+1,J)/(SR(1)+BX)
1142.....DO 33 I=1,N2
1143.....GB(I,1)=GB(I,3)
1144.....GC(I,1)=-GC(I,3)
1145.....33-GB(I,1)=GB(I,3)
1146.....CALL QACOPY(B1(1,1,2),GC,NS)
1147.....CALL QACOPY(B2(1,1,2),GB,NS)
1148.....CALL QACOPY(C3(1,1,2),GB,NS)
1149.....RETURN
1150.....END
1151
1152
1153
1154
1155
1156
1157.....SUBROUTINE BMDRY
1158 C..... THIS SUBROUTINE CALCULATES PSI ON THE
1159 C..... PLASMA-VACUUM INTERFACE USING THE CONTINUITY OF PSI AND
1160 C..... ITS NORMAL DERIVATIVE; IT IS CALLED BY "FLX"
1161 C..... LCM (VARIABLES), (LCMICGG), (COEFAD), (CONTOUR), (WE2M2)
1162 C..... COMMON/VARIABLES/PSI(IN2,JM1P1,3),B1(IN2,JM1P1,2),
1163 C..... B2(IN2,JM1P1,2),B3(IN2,JM1P1,2),XI(IN2,JM1P1,2),Z1(IN2,JM1P1,2),
1164 C..... I02(IN2,JM1P1,2),C3(IN2,JM1P1,2),
1165 C..... 2T(IN2,JM1P1,2),W(4,IN2,JM1P1,3),F(4,IN2,JM1P1,3),

```



```

1166.....GEI(IN2,JM1P1,2),CN(IN2,JM1P1,2)
1167.....COMMON/CONST2/IS,IB,IE,IB,IP,JS,JB,JE,JB,JP,BX,BY,BATA,BATB
1168.....COMMON/MESH/RR(NT),ZZ(MT),BELRR(NT-1),BELZZ(MT-1),
1169.....ISR(IN2),SZ(JM1P1),BELR(IN2-1),BELZ(JM1P1-1)
1170.....COMMON/CROUTV/AA(IN2,3),B(IN2),X1(IN2)
1171.....COMMON/CONST/N1,N2,N3,M1,M2,N1P1,N1P2,N2P1,N2P2,N2M1,
1172.....N1PN2,N1PN2P,N1PN2M,M1P1,M1M1,M1P2,M2P1,M2M1,M2P2,NN,NM1,MM,MMM1
1173.....COMMON/CUR/NCUR,LOGCUR(35),VALCUR(35)
1174.....COMMON/DIM/RSMA,RB,BB1,RLNGTH,ZLNGTH
1175.....COMMON/COEFAD/COEFR1(IN2-1,JM1P1-1),COEFR2(IN2-1,JM1P1-1),
1176.....2COEFR3(IN2-1,JM1P1-1),COEFZ1(IN2-1,JM1P1-1),
1177.....3COEFZ2(IN2-1,JM1P1-1),COEFZ3(IN2-1,JM1P1-1)
1178.....COMMON/CONTOUR/FLUXRZ(NT,MT),RAXIS(NT),ZAXIS(MT)
1179.....COMMON/VAR2/FR,FK,CV,C,Q,HU,UN,BT,TMAX,PTIME1,PTIME2,TIME,
1180.....1-TDIS1,TDIS2,
1181.....1BMIN,SGO,CDE,MW,MP,MC,NCURY,NCHECK,BTOR,DN,TEMP,
1182.....2CUR1MAX,CUR2MAX,CUR3MAX,CURTOR,RSPINCH,RBPINCH
1183.....COMMON/WSEM/S(N,(-4,4))
1184.....COMMON/WLCM2/GA(IN2,JM1P1),GB(IN2,JM1P1),GC(IN2,JM1P1),
1185.....1GB(IN2,JM1P1),GE(IN2,JM1P1),GF(IN2,JM1P1),VSEM1(4,IN2,JM1P1),
1186.....2VSEM2(4,IN2,JM1P1),APSI(NT,MT)
1187.....COMMON/LCMICG/MTRX(N,(-4,4)),LU(N,(-4,4)),X(N),Y(N)
1188.....REAL MTRX,LU
1189.....OPTIMIZE
1190.....C....I=N1+1,M2+2,LE,J,LE,MM
1191.....I=N1P1
1192.....A=(RR(I+2)-RR(I))/(RR(I+1)-RR(I))
1193.....B=(RR(I)-RR(I-2))/(RR(I)-RR(I-1))
1194.....DO 10 J=M2P2,MMM1
1195.....APSI(I,J)=-(A*BELRR(I-1)+(APSI(I-2,J)-B*B*APSI(I-1,J))/(B-1.0)+
1196.....1.....B*BELRR(I-1)+(APSI(I+2,J)-A*A*APSI(I+1,J))/(A-1.0)
1197.....2.....)/(A+(B+1.0)*BELRR(I)+B*(A+1.0)*BELRR(I-1))
1198.....10 PSI(N2,MM-J+1,3)=APSI(I,J)
1199.....APSI(I,MM)=APSI(I,MM-2)
1200.....PSI(N2,1,3)=PSI(N2,3,3)
1201.....C....J=M2+1,N1+1,LE,I,LE,NM1
1202.....J=M2P1
1203.....A=(ZZ(J+2)-ZZ(J))/(ZZ(J+1)-ZZ(J))
1204.....B=(ZZ(J)-ZZ(J-2))/(ZZ(J)-ZZ(J-1))
1205.....DO 20 I=N1P1,NM1
1206.....APSI(I,J)=-(A*BELZZ(J-1)+(APSI(I,J-2)-B*B*APSI(I,J-1))/(B-1.0)+

```

187

```

1246.....TEST=SQ♦TOL
1247.....IF(NP.EQ.1) GO TO JUMP5
1248.....SQ=SQ-S(NP,-1)♦S(NP-1,1)
1249.....IBM=MIN0(4,NP-IR3-1)
1250.....DO LOOP4 IB=2,IBM
1251..LOOP4 SQ=SQ-S(NP,-IB)♦S(NP-IR3-IB,IB)
1252..JUMP5 IF (ABS(SQ).GT.ABS(TEST)) GO TO JUMP6
1253..C....PIVOT ELEMENT TOO SMALL
1254.....SQ=TEST
1255.....LUERR=LUERR+1
1256.....IF(SQ.NE.0.) GO TO JUMP6
1257.....SQ=1.
1258..JUMP6 CONTINUE
1259.....ELPPI=1/SQ
1260.....S(NP,0)=SQ
1261.....IF(NP.EQ.N) GO TO LOOP30
1262..C....COMPLETE SOLUTION OF LOWER TRIANGULAR MATRIX:
1263..C....MOVE DOWN PIVOT COLUMN:
1264..C....SOLVE FOR S(NP+1,-1)
1265.....SQ=S(NP+1,-1)
1266.....IF(NP-IR3-2.LT.1) GO TO JUMP10
1267.....SQ=SQ-S(NP+1,-3)♦S(NP-IR3-2,2)
1268.....IF(NP-IR3-3.LT.1) GO TO JUMP10
1269.....SQ=SQ-S(NP+1,-4)♦S(NP-IR3-3,3)
1270..JUMP10 S(NP+1,-1)=SQ
1271.....IF(NP+IR3+2.GT.N) GO TO JUMP16
1272..C....SOLVE FOR S(NP+IR3+2,-2)
1273.....SQ=S(NP+IR3+2,-2)
1274.....IF(NP.EQ.1) GO TO JUMP12
1275.....SQ=SQ-S(NP+IR3+2,-3)♦S(NP-1,1)
1276..JUMP12 S(NP+IR3+2,-2)=SQ
1277.....IF(NP+IR3+3.GT.N) GO TO JUMP16
1278..C....SOLVE FOR S(NP+IR3+3,-3)
1279.....SQ=S(NP+IR3+3,-3)
1280.....IF(NP.EQ.1) GO TO JUMP14
1281.....SQ=SQ-S(NP+IR3+3,-4)♦S(NP-1,1)
1282..JUMP14 S(NP+IR3+3,-3)=SQ
1283..JUMP16 CONTINUE
1284..C....COMPUTE SOLUTION OF UPPER TRIANGULAR MATRIX:
1285..C....MOVE RIGHT ALONG PIVOT ROW:

```

```

1286 C..... SOLVE FOR S(NP,1)
1287 ..... SQ=S(NP,1)
1288 ..... IF (NP-IR3-2.LT.1) GO TO JUMP20
1289 ..... SQ=SQ-S(NP,-2)*S(NP-IR3-2,3)
1290 ..... IF (NP-IR3-3.LT.1) GO TO JUMP20
1291 ..... SQ=SQ-S(NP,-3)*S(NP-IR3-3,4)
1292 JUMP20 S(NP,1)=SQ*ELPPI
1293 ..... IF (NP+2+IR3.GT.N) GO TO JUMP26
1294 C..... SOLVE FOR S(NP,2)
1295 ..... SQ=S(NP,2)
1296 ..... IF (NP.LE.1) GO TO JUMP22
1297 ..... SQ=SQ-S(NP,-1)*S(NP-1,3)
1298 JUMP22 S(NP,2)=SQ*ELPPI
1299 ..... IF (NP+3+IR3.GT.N) GO TO JUMP26
1300 C..... SOLVE FOR S(NP,3)
1301 ..... SQ=S(NP,3)
1302 ..... IF (NP.LE.1) GO TO JUMP24
1303 ..... SQ=SQ-S(NP,-1)*S(NP-1,4)
1304 JUMP24 S(NP,3)=SQ*ELPPI
1305 ..... IF (NP+4+IR3.GT.N) GO TO JUMP26
1306 C..... STORE S(NP,4)
1307 ..... S(NP,4)=S(NP,4)*ELPPI
1308 JUMP26 CONTINUE
1309 LOOP30 CONTINUE $$$ L-U DECOMPOSITION LOOP *****
1310 C..... SET S(I,0)=1/S(I,0)
1311 ..... SQ=1
1312 ..... CALL QVDIV2(S(1,0),SQ,S(1,0),N)
1313 ..... CALL QACOPY(LU,S,9*N)
1314 ..... RETURN
1315
1316 C..... *****
1317 C..... *****
1318 C..... ENTER HERE TO DO CONJUGATE GRADIENT ITERATION.
1319
1320 ..... ENTRY CGITER(KFLAG,BXNORM,RNORM)
1321 ..... IF (KFLAG.EQ.1) GO TO JUMP40
1322 ..... KFLAG=1
1323 C..... INITIALIZE FOR 1ST CG ITERATION
1324 ..... CALL QACOPY(P,X,N)
1325 ..... CALL QACOPY(R,Y,N)

```

```

1326 C.... YNORM=ABS(SUM(Y(I))
1327 ..... YNORM=0. $ CALL QYDOT(YNORM,R,R,N) $ YNORM=SQRTI(YNORM)
1328 ..... IF(YNORM.EQ.0.0) YNORM=1.E+70
1329 ..... A=1.
1330 ..... RDOT=1.E90
1331 JUMP40 CONTINUE $$$ ITERATION STARTS .....
1332 C.... .....
1333 C.... EVALUATE RESIDUAL, R=R-A*M*P=Y-M*X
1334 C.... M*P EVALUATED AND STORED IN S1
1335 ..... CALL QYSET(0.,S1,N)
1336
1337 ..... DO LOOP50 IB=-3,3
1338 ..... IF(IABS(IB).EQ.2) GO TO LOOP50
1339 ..... IIB=IB
1340 ..... IF(IB.LT.-1) IIB=IIB-IR3
1341 ..... IF(IB.GT.1) IIB=IIB+IR3
1342 ..... I1=MAX0(1,1-IIB) $ I2=MIN0(N,N-IIB)
1343 ..... CALL QACOPY(S2,MTRX(1,IB),N)
1344 ..... CALL QMA0(S1(I1),S2(I1),P(I1+IIB),S1(I1),I2-I1+1)
1345 LOOP50 CONTINUE
1346 ..... CALL QMR1(R,R,S1,A,N) $$$ R(I)=R(I)-S1(I)*A
1347 C.... COMPUTE RNORM=ABS(SUM(R(I))/YNORM
1348 ..... RNORM=0. $ CALL QYDOT(RNORM,R,R,N)
1349 ..... RNORM=SQRTI(RNORM)/YNORM
1350 C.... .....
1351 C.... COMPUTE P=(UT*U) + (-1)*MT*(L*LT) + (-1)*R+B*P
1352 C.... S1=L + (-1)*R VIA FORWARD SUBSTITUTION
1353 ..... CALL QACOPY(EL,LU,N*5)
1354 ..... S1(1)=EL(1,0)*R(1)
1355 ..... DO LOOP54 I=2,IR-1
1356 LOOP54 S1(I)=EL(I,0)*(R(I)-EL(I,-1)*S1(I-1))
1357 ..... I=IR
1358 ..... S1(I)=EL(I,0)*(R(I)-EL(I,-1)*S1(I-1)-
1359 ..... 1*EL(I,-2)*S1(I))
1360 ..... I=IR+1
1361 ..... S1(I)=EL(I,0)*(R(I)-EL(I,-1)*S1(I-1)-
1362 ..... 2*EL(I,-2)*S1(I)-EL(I,-3)*S1(I))
1363 ..... DO LOOP55 I=IR+2,N
1364 LOOP55 S1(I)=EL(I,0)*(R(I)-EL(I,-1)*S1(I-1)-
1365 ..... 3*EL(I,-2)*S1(I-IR+1)-EL(I,-3)*S1(I-IR))
1366 C.... COMPUTE RDOT=(S1,S1) .....

```

```

1367.....RDOTL=RDOT
1368.....RDOT=0...$...CALL QVBDT(RDOT,S1,S1,N)
1369...C....S2=LT+(-1)*S1...VIA BACKWARD SUBSTITUTION
1370.....S2(N)=EL(N,0)*S1(N)
1371.....DO LOOP56 I=N-1,N-IR+2,-1
1372...LOOP56 S2(I)=EL(I,0)*(S1(I)-EL(I+1,-1)*S2(I+1))
1373.....I=N-IR+1
1374.....S2(I)=EL(I,0)*(S1(I)-EL(I+1,-1)*S2(I+1)-
1375.....5*EL(N,-2)*S2(N))
1376.....I=N-IR
1377.....S2(I)=EL(I,0)*(S1(I)-EL(I+1,-1)*S2(I+1)-
1378.....6*EL(N-1,-2)*S2(N-1)-EL(N,-3)*S2(N))
1379.....DO LOOP57 I=N-IR-1,1,-1
1380...LOOP57 S2(I)=EL(I,0)*(S1(I)-EL(I+1,-1)*S2(I+1)-
1381.....9*EL(I+IR-1,-2)*S2(I+IR-1)-EL(I+IR,-3)*S2(I+IR))
1382...C....S1=MT*S2
1383.....CALL QYSET(0.,S1,N)
1384
1385.....DO LOOP60 IB=-3,3
1386.....IF (IABS(IB).EQ.2) GO TO LOOP60
1387.....IIB=IB
1388.....IF (IB.LT.-1) IIB=IIB-IR3
1389.....IF (IB.GT.1) IIB=IIB+IR3
1390.....I1=MAX0(1,1+IIB)...$...I2=MIN0(N,N+IIB)
1391.....CALL QACOPY(S,MTRX(1,IB),N)
1392.....CALL QMA0(S1(I1),S(I1-IIB),S2(I1-IIB),S1(I1),I2-I1+1)
1393...LOOP60 CONTINUE
1394...C....S2=UT+(-1)*S1
1395.....CALL QACOPY(U(1,1),LU(1,1),3*N)
1396.....S2(1)=S1(1)
1397.....DO LOOP61 I=2,IR-1
1398...LOOP61 S2(I)=S1(I)-U(I-1,1)*S2(I-1)
1399.....I=IR
1400.....S2(I)=S1(I)-U(I-1,1)*S2(I-1)-
1401.....2*U(1,1)*S2(1)
1402.....I=IR+1
1403.....S2(I)=S1(I)-U(I-1,1)*S2(I-1)-
1404.....3*U(2,1)*S2(2)-U(1,2)*S2(1)
1405.....DO LOOP62 I=IR+2,N

```

192

```

1446 C:.... TEST DX FOR SMALLNESS
1447 ..... DXNORM=0.0 $ CALL QYDOT (DXNORM,P,P,N)
1448 ..... XNORM=0.0 $ CALL QYDOT (XNORM,S,S,N)
1449 ..... DXNORM=SQRT (DXNORM**2+XNORM**2)
1450 ..... RETURN
1451 ..... END
1452
1453
1454
1455
1456
1457
1458 ..... SUBROUTINE INITIAL (CONBUC,RZERO,PNGHT,TAU0,TAU1,IC1B,IC1E,
1459 ..... 1JC2B,JC2E,JC3E,G1R1,G1R2,G2Z1,G2Z2,G3Z2)
1460 C:.... THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
1461 C:.... TO DO VARIOUS INITIALIZATIONS. IT ALSO SETS UP THE INITIAL
1462 C:.... PINCH WITH A TOROIDAL CURRENT ETC.
1463 ..... LCM (VARIABLES), (LCMICCG), (CDEFAB), (CONTOUR), (WLEM2)
1464 ..... COMMON/VARIABLES/PSI (IN2,JM1P1,3), B1 (IN2,JM1P1,2),
1465 ..... 1B2 (IN2,JM1P1,2), B3 (IN2,JM1P1,2), XI (IN2,JM1P1,2), S1 (IN2,JM1P1,2),
1466 ..... 1C2 (IN2,JM1P1,2), C3 (IN2,JM1P1,2),
1467 ..... 2T (IN2,JM1P1,2), W (4,IN2,JM1P1,3), F (4,IN2,JM1P1,3),
1468 ..... 3EI (IN2,JM1P1,2), CN (IN2,JM1P1,2)
1469 ..... COMMON/CONST2/IS, IB, IE, ID, IP, JS, JB, JE, JB, JP, DX, DY, BATA, BATB
1470 ..... COMMON /MESH/ RR (NT), ZZ (MT), DELRR (NT-1), DELZZ (MT-1),
1471 ..... 1SR (IN2), SZ (JM1P1), DELR (IN2-1), DELZ (JM1P1-1)
1472 ..... COMMON/CROUTY/ RA (IN2,3), B (IN2), X1 (IN2)
1473 ..... COMMON/CONST/ N1,N2,N3,M1,M2,N1P1,N1P2,N2P1,N2P2,N2M1,
1474 ..... 1N1PN2,N1PN2P,N1PN2M,M1P1,M1M1,M1P2,M2P1,M2M1,M2P2,NN,NNM1,MM,MMM1
1475 ..... COMMON/CUR/ NEUR,LECCUR (35), VALCUR (35)
1476 ..... COMMON/DIM/RSMA,RB,BB1,RLNGTH,ZLNGTH
1477 ..... COMMON/CDEFAB/CDEFR1 (IN2-1,JM1P1-1),CDEFR2 (IN2-1,JM1P1-1),
1478 ..... 2CDEFR3 (IN2-1,JM1P1-1),CDEFZ1 (IN2-1,JM1P1-1),
1479 ..... 3CDEFZ2 (IN2-1,JM1P1-1),CDEFZ3 (IN2-1,JM1P1-1)
1480 ..... COMMON/CONTOUR/ FLUXRZ (NT,MT),RAXIS (NT),ZAXIS (MT)
1481 ..... COMMON/VAR2/FR,FK,CV,C,Q,HU,UN,BT,TMAX,PTIME1,PTIME2,TIME,
1482 ..... 1TBIS1,TBIS2,
1483 ..... 1BMIN,SGO,CDE,MW,MP,MC,NEURY,NCHECK,BTOR,BN,TEMP,
1484 ..... 2CUR1MAX,CUR2MAX,CUR3MAX,CURTOR,RSPINCH,RBPINCH
1485 ..... COMMON /WSEM/ S (N, (-4,4))

```



```

1486.....COMMON/ULECM2/ GA(IN2,JM1P1),GB(IN2,JM1P1),GC(IN2,JM1P1),
1487.....1GB(IN2,JM1P1),GE(IN2,JM1P1),GF(IN2,JM1P1),VSCM1(4,IN2,JM1P1),
1488.....2VSCM2(4,IN2,JM1P1),APSI(MT,MT)
1489.....COMMON/LEMICCG/ MTRX(N,(-4,4)),LU(N,(-4,4)),X(N),Y(N)
1490.....REAL MTRX,LU
1491.....OPTIMIZE
1492.....CALL KEEP80(1,2)
1493.....CALL DD80ID(6HTORMAC,1)
1494.....WRITE(3,7) N1,N2,N3,M1,M2,RB,RSMA,BB1
1495.....7 FORMAT(//5X,"N1...=",I4/5X,"N2...=",I4/5X,"N3...=",
1496.....1 I4/5X,"M1...=",I4/5X,"M2...=",I4/5X,
1497.....2 "RB...=",E12.3/5X,"RSMA=",E12.3/5X,"BB1=",E12.3)
1498.....NOW DEFINE SOME CONSTANTS
1499.....N1P1=N1+1
1500.....N1P2=N1+2
1501.....N2P1=N2+1
1502.....N2P2=N2+2
1503.....N2M1=N2-1
1504.....N1PN2=N1+N2
1505.....N1PN2P=N1+N2+1
1506.....N1PN2M=N1+N2-1
1507.....M1P1=M1+1
1508.....M1M1=M1-1
1509.....M2P1=M2+1
1510.....M2M1=M2-1
1511.....M1P2=M1+2
1512.....M2P2=M2+2
1513.....NN=N1+N2+N3
1514.....MM=M1+M2+1
1515.....NNM1=NN-1
1516.....MMM1=MM-1
1517.....IB=2
1518.....JB=2
1519.....IS=IB-1
1520.....JS=JB-1
1521.....IB=N2
1522.....JB=M1P1
1523.....JE=JB-1
1524.....IE=IB-1
1525.....IP=IE-IS

```

```

1526 .....JP=JE-JS
1527 .....UN=BTOR/SORT(4.♦3.14159♦BN)
1528 .....BT=BT♦UN/RSMA
1529 .....TMAX=TMAX♦UN/RSMA
1530 .....PTIME1=PTIME1♦UN/RSMA
1531 .....PTIME2=PTIME2♦UN/RSMA
1532 .....TDIS1=TDIS1♦UN/RSMA
1533 .....TDIS2=TDIS2♦UN/RSMA
1534 .....TAU0=TAU0♦UN/RSMA
1535 .....TAU1=TAU1♦UN/RSMA
1536 .....BMIN=BMIN/BTOR
1537 .....SGO=CONBUC♦(TEMP/11600.)♦♦1.5
1538 .....COE=C♦C♦BT/(4.♦3.14159♦RSMA♦UN♦SGO)
1539 C.....CALCULATE SR(I),SZ(J),DELR(I),DELZ(J),RR(I),ZZ(J),DELRR(I),
1540 C.....DELZZ(J)
1541 .....Z1=BB1/RSMA
1542 .....SR(1)=(RB-RSMA)/RSMA
1543 .....DEL=2.♦RSMA/((N2-1)♦RSMA)
1544 .....DO 25 I=2,N2
1545 ... 25 SR(I)=SR(I-1)+DEL
1546 .....DO 26 I=1,N2M1
1547 ... 26 DELR(I)=SR(I+1)-SR(I)
1548 .....SZ(2)=0.
1549 .....DEL=BB1/((M1-1)♦RSMA)
1550 .....DO 27 J=3,M1P1
1551 ... 27 SZ(J)=SZ(J-1)+DEL
1552 .....SZ(1)=-SZ(3)
1553 .....DO 28 J=1,M1
1554 ... 28 DELZ(J)=SZ(J+1)-SZ(J)
1555 .....DX=DELR(1)
1556 .....BATA=BT/(2.♦DX)
1557 .....BY=DELZ(1)
1558 .....BATB=BT/(2.♦BY)
1559 C.....NOTE THAT WE ARE ASSUMING DX & BY ARE CONSTANTS.
1560 .....RHO1=1./(SR(N2)+RLNGTH)
1561 .....RHO2=1./(SR(1)+RLNGTH)
1562 .....ZETA1=1./(Z1+ZLNGTH)
1563 .....ZETA2=1./ZLNGTH
1564 .....RR(1)=0.
1565 .....DEL=(1.-0.925)/(1.-0.925♦♦N1)♦RHO1

```

```

1566.....DO 16 I=2,N1
1567.....RR(I)=RR(I-1)+DEL
1568.....16 DEL=0.925*DEL
1569.....DO 17 I=1,N2
1570.....17 RR(I+N1)=1./(SR(N2+1-I)+RLNGTH)
1571.....DO 19 I=1,NNM1
1572.....19 DELRR(I)=RR(I+1)-RR(I)
1573.....ZZ(1)=0.
1574.....DEL=(1.-0.925)/(1.-0.925**M2)*ZETA1
1575.....DO 20 J=2,M2
1576.....ZZ(J)=ZZ(J-1)+DEL
1577.....20 DEL=0.925*DEL
1578.....DO 22 J=1,M1P1
1579.....22 ZZ(M2+J)=1./(SZ(M1+2-J)+ZLNGTH)
1580.....DO 24 J=1,MMM1
1581.....24 DELZZ(J)=ZZ(J+1)-ZZ(J)
1582.....DO 803 I=1,NNM1
1583.....803 RAXIS(I)=(1.-RLNGTH*RR(NN+1-I))/RR(NN+1-I)
1584.....RAXIS(NN)=9.99E+300
1585.....DO 802 J=1,MMM1
1586.....802 ZAXIS(J)=(1.-ZLNGTH*ZZ(MM+1-J))/ZZ(MM+1-J)
1587.....ZAXIS(MM)=9.99E+300
1588.....WRITE(3,34)
1589.....34 FORMAT(/5X,"SR:..")/
1590.....WRITE(3,30) (SR(I),I=1,N2)
1591.....WRITE(3,35)
1592.....35 FORMAT(/5X,"SZ:..")/
1593.....WRITE(3,30) (SZ(J),J=1,M1P1)
1594.....WRITE(3,36)
1595.....36 FORMAT(/5X,"DELRR:..")/
1596.....WRITE(3,30) (DELRR(I),I=1,N2M1)
1597.....WRITE(3,37)
1598.....37 FORMAT(/5X,"DELZZ:..")/
1599.....WRITE(3,30) (DELZZ(J),J=1,M1)
1600.....WRITE(3,38)
1601.....38 FORMAT(/5X,"RR:..")/
1602.....WRITE(3,30) (RR(I),I=1,NN)
1603.....WRITE(3,39)
1604.....39 FORMAT(/5X,"ZZ:..")/
1605.....WRITE(3,30) (ZZ(J),J=1,MM)

```

```

1606.....WRITE(3,40)
1607....40 FORMAT(/5X,"DELRR:....AND DELZZ:.."/)
1608.....WRITE(3,30) (DELRR(I), I=1,NNM1)
1609.....WRITE(3,30) (DELZZ(J), J=1,MMM1)
1610....30 FORMAT(10E12.4)
1611.....WRITE(3,804)
1612...804 FORMAT(/" R-AXIS"/)
1613.....WRITE(3,30) (RAXIS(I), I=1,NN)
1614.....WRITE(3,805)
1615...805 FORMAT(/" Z-AXIS"/)
1616.....WRITE(3,30) (ZAXIS(J), J=1,MM)
1617....2 FORMAT(20I5)
1618 COM.....DETERMINE IC1B, IC1E, JC2B, JC2E, JC3E
1619.....DO 100 I=1, N2
1620.....IF (SR(I):LT:C1R1) GO TO 100
1621.....IC1B=I
1622.....GO TO 101
1623...100 CONTINUE
1624...101 DO 102 I=1, N2
1625.....IF (SR(I):LT:C1R2) GO TO 102
1626.....IC1E=I-1
1627.....GO TO 103
1628...102 CONTINUE
1629...103 DO 104 J=2, M1P1
1630.....IF (SZ(J):LT:C3Z2) GO TO 104
1631.....JC3E=J-1
1632.....GO TO 105
1633...104 CONTINUE
1634...105 DO 106 J=2, M1P1
1635.....IF (SZ(J):LT:C2Z1) GO TO 106
1636.....JC2B=J
1637.....GO TO 107
1638...106 CONTINUE
1639...107 DO 108 J=2, M1P1
1640.....IF (SZ(J):LT:C2Z2) GO TO 108
1641.....JC2E=J-1
1642.....GO TO 109
1643...108 CONTINUE
1644 COM.....INITIALIZE LCCUR(L, L=1, NCUR)
1645...109 NCUR=(IC1E-IC1B+1)+JC2E-JC2B+1+(JC3E-2+1)

```

```

1646.....L=L+1
1647.....DO 110 I=IC1B, IC1E
1648.....L=L+1
1649.....110 LOGCUR(L)=100*(N1P1-2-I+1)+M2
1650.....DO 112 J=JC2B, JC2E
1651.....L=L+1
1652.....112 LOGCUR(L)=100*(N1+(MM-J+1))
1653.....DO 114 J=2, JC3E
1654.....L=L+1
1655.....114 LOGCUR(L)=100*(N1+(MM-J+1))
1656.....213 CALL QVSET(0, VALCUR, NCUR)
1657.....CUR1MAX=CUR1MAX+4.*3.14159/(C+RSMA+BTOR)
1658.....3/((SR(IC1E+1)-SR(IC1B))*+(ZAXIS(M1P1+2)-ZAXIS(M1P1))/2.)
1659.....CUR2MAX=CUR2MAX+4.*3.14159/(C+RSMA+BTOR)
1660.....2/((SZ(JC2E+1)-SZ(JC2B))*+(RAXIS(N2+2)-RAXIS(N2))/2.)
1661.....CUR3MAX=CUR3MAX+4.*3.14159/(C+RSMA+BTOR)
1662.....1/((SZ(JC3E)+SZ(JC3E+1))*+(RAXIS(N2+2)-RAXIS(N2))/2.)
1663.....RSPINCH=RSPINCH/RSMA
1664.....RBPINCH=RBPINCH/RSMA
1665.....RZERO=RZERO/RSMA
1666.....AREAP=3.14159*RSPINCH**2
1667.....CURTOR=CURTOR+4.*3.14159/(C+RSMA+BTOR)/AREAP
1668.....DO 5 I=1, NM
1669.....DO 5 J=1, MM
1670.....5 APSI(I, J)=0.
1671.....DO 6 K=1, 2
1672.....DO 6 I=1, N2
1673.....DO 6 J=1, M1P1
1674.....XI(I, J, K)=BMIN/Q
1675.....PSI(I, J, K)=0.
1676.....PSI(I, J, K+1)=0.
1677.....EI(I, J, K)=TEMP*CV/(UN+UN)
1678.....EN(I, J, K)=2.*8E+06/(DN+RSMA+UN+CV)
1679.....T(I, J, K)=1.
1680.....B1(I, J, K)=0.
1681.....B2(I, J, K)=0.
1682.....B3(I, J, K)=XI(I, J, K)/SR(I)
1683.....C1(I, J, K)=0.
1684.....C2(I, J, K)=0.
1685.....C3(I, J, K)=0.

```

```

1686.....W(1,I,J,K)=1.0/2.5
1687.....W(2,I,J,K)=0.
1688.....W(3,I,J,K)=0.
1689.....W(4,I,J,K)=0.
1690.....W(1,I,J,K+1)=1.0/2.5
1691.....W(2,I,J,K+1)=0.
1692.....W(3,I,J,K+1)=0.
1693.....W(4,I,J,K+1)=0.
1694.....6 CONTINUE
1695.....K=1
1696.....DO 42 I=1,N2
1697.....DO 41 J=2,M1P1
1698.....RAD=SQRT((SR(I)-RBPINCH)**2+SZ(J)**2)
1699.....IF (RAD.GT.RSPINCH) GO TO 44
1700.....PSI(I,J,K)=PSI(I,J,K)-RBPINCH/4.0*CURTOR*(RAD+RAD
1701.....1-.....(RSPINCH**2)*(1.0-2.0*LOGI(RSPINCH/RZERO)))
1702.....W(1,I,J,K)=2.0
1703.....W(1,I,J,K+1)=2.0
1704.....EI(I,J,K)=PNEHT*EI(I,J,K)
1705.....EI(I,J,K+1)=EI(I,J,K)
1706.....GO TO 41
1707....44 IF (RAD.GT.RZERO) GO TO 41
1708.....PSI(I,J,K)=PSI(I,J,K)-RBPINCH/2.0*CURTOR
1709.....1.....*(RSPINCH**2)*LOGI(RAD/RZERO)
1710....41 CONTINUE
1711.....PSI(I,1,K)=PSI(I,3,K)
1712.....W(1,I,1,K)=W(1,I,3,K)
1713.....W(1,I,1,K+1)=W(1,I,3,K+1)
1714.....EI(I,1,K)=EI(I,3,K)
1715.....EI(I,1,K+1)=EI(I,3,K+1)
1716....42 CONTINUE
1717.....CALL GGCDEF
1718.....CALL QYSET(0.,X,N)
1719.....CALL QYSET(0.,Y,N)
1720.....RETURN
1721.....END
1722
1723
1724

```

```

1725 ..... SUBROUTINE CLPLOT (IDN, IPSI, IDNTP, JJZ1, IFIEL, JJZ2, M,
1726 ..... 2RSMA, UN, TIME, NCURY, BTOR)
1727 C..... THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM TO DO
1728 C..... VARIOUS GRAPHING CHORES.
1729 C..... IT MAKES EXTENSIVE USE OF THE "TV80" LIBRARY AT MFECC.
1730 ..... LCM (VARIABLES), (LCMICCG), (CDEFAD), (CONTOUR), (WLCM2)
1731 ..... COMMON/VARIABLES/PSI (IN2, JM1P1, 3), B1 (IN2, JM1P1, 2),
1732 ..... B2 (IN2, JM1P1, 2), B3 (IN2, JM1P1, 2), XI (IN2, JM1P1, 2), C1 (IN2, JM1P1, 2),
1733 ..... C2 (IN2, JM1P1, 2), C3 (IN2, JM1P1, 2),
1734 ..... 2T (IN2, JM1P1, 2), W (4, IN2, JM1P1, 3), F (4, IN2, JM1P1, 3),
1735 ..... 3EI (IN2, JM1P1, 2), CN (IN2, JM1P1, 2)
1736 ..... COMMON/CONST2/IS, IB, IE, ID, IP, JS, JB, JE, JD, JP, DX, DY, BATA, BATB
1737 ..... COMMON/MESH/RR (NT), ZZ (MT), DELRR (NT-1), DELZZ (MT-1),
1738 ..... 1SR (IN2), SZ (JM1P1), DELR (IN2-1), DELZ (JM1P1-1)
1739 ..... COMMON/CONST/ N1, N2, N3, M1, M2, N1P1, N1P2, N2P1, N2P2, N2M1,
1740 ..... 1N1PN2, N1PN2P, N1PN2M, M1P1, M1M1, M1P2, M2P1, M2M1, M2P2, NN, NNM1, MM, MMM1
1741 ..... COMMON/CONTOUR/ FLUXRZ (NT, MT), RAXIS (NT), ZAXIS (MT)
1742 ..... DIMENSION C (IN2), X (IN2), XX (IN2, JM1P1), YY (IN2, JM1P1),
1743 ..... 2PICT (IN2, JM1P1), CVAL (35)
1744 ..... SIME=RSMA/UN*TIME
1745 ..... CALL DBERS (-1)
1746 ..... DO 2 I=1, N2
1747 ..... 2 X(I)=SR(I)*RSMA
1748 ..... IF (IPSI.NE.1) GO TO 100
1749 C..... DRAW PSI CONTOURS.
1750 ..... DO 50 II=1, 2
1751 ..... ISW=II-1
1752 ..... CALL CONTR (1, N2+2, 2, M1P1+2, NCURY, ISW, CVAL, M)
1753 ..... CALL SETCH (15, 31, 0, 0, 2)
1754 ..... WRITE (100, 10) M, SIME
1755 ..... 10 FORMAT ("M...=", I4/"TIME=", E9.2, " SEC."/"POLOIDAL FLUX CONTOURS")
1756 COM... CALL SETCH (52, 23, 0, 0, 2)
1757 COM... DO 51 I=1, NCURY
1758 COM... CVAL (I)=CVAL (I)*BTOR*RSMA**2
1759 COM51 WRITE (100, 52) CVAL (I)
1760 COM52 FORMAT (E11.2)
1761 ..... 50 CONTINUE
1762 ..... DO 5 I=1, N2
1763 ..... DO 5 J=2, M1P1
1764 ..... 5 FLUXRZ (I, J)=SQRT (B1 (I, J, 1)**2+B2 (I, J, 1)**2+B3 (I, J, 1)**2)

```

```

1765.....ISW=0...$...CALL CONTR(1,N2,2,M1P1,NEURY,ISW,CVAL,M)
1766.....CALL SETCH(15.,32.,0,0,2)
1767.....WRITE(100,6) M,SIME
1768.....6 FORMAT("M...=",I4/"TIME=",E9.2,"SEC:"/
1769.....1"CONSTANT ABS(B) CONTOURS")
1770...100 IF (IDN.NE.1) GO TO 200
1771...6.....PLOT DENSITY CONTOURS
1772.....DO 102 I=1,N2
1773.....DO 102 J=2,M1P1
1774.....PICT(I,J-1)=W(1,I,J,3)
1775...102 FLUXRZ(I,J)=W(1,I,J,3)
1776.....PEAK=-9.9E+99
1777.....DO 115 I=1,N2
1778.....DO 115 J=1,M1
1779.....IF (PICT(I,J).GT.PEAK) PEAK=PICT(I,J)
1780...115 CONTINUE
1781.....ISW=0
1782.....CALL CONTR(1,N2,2,M1P1,NEURY,ISW,CVAL,M)
1783.....CALL SETCH(15.,32.,0,0,2)
1784.....WRITE(100,104) M,SIME
1785...104 FORMAT("M...=",I4/"TIME=",E9.2,"SEC:"/"DENSITY CONTOURS")
1786.....CALL FRAME
1787.....RATIO=(X(N2)-X(1))/RSMA/(SZ(M1P1)-SZ(1))
1788.....CALL MAPS(X(1),X(N2),SZ(1)♦RSMA,SZ(M1P1)♦RSMA,
1789.....1 0.10, 0.99, 0.05, 0.05+0.89/RATIO)
1790.....DO 501 I=2,N2M1
1791.....DO 501 J=2,M1
1792.....X0=SR(I)♦RSMA
1793.....Y0=SZ(J)♦RSMA
1794.....DU=DX♦RSMA+UN♦W(2,I,J,2)/W(1,I,J,2)/5.0E+6
1795.....DV=DY♦RSMA+VN♦W(3,I,J,2)/W(1,I,J,2)/5.0E+6
1796.....CALL PLOTV(X0,Y0,X0+DU,Y0+DV)
1797...501 CONTINUE
1798.....CALL SETCH(15., 31., 0., 0., 2)
1799.....WRITE(100,503) M,SIME
1800...503 FORMAT("M...=",I4/"TIME=",E9.2,"SEC:"/"POLOIDAL FLOW")
1801.....XW=(RAXIS(N2)-RAXIS(1))♦RSMA
1802.....XW=XW/XW/1000.
1803.....YW=(ZAXIS(M1P1)-ZAXIS(2))♦RSMA
1804.....LHIDE=2

```



```

1805.....LBOX=3
1806...110 FORMAT("M...=",I4,3X,"TIME=",E9.2,"SEC:"/"DENSITY"
1807.....1,2X,"(MAXIMUM=",E9.2,"")")
1808.....XVIEW=+4:RSMA
1809.....YVIEW=-3:RSMA
1810.....ZVIEW=+4:RSMA
1811.....CALL FRAME
1812.....CALL PICTURE(PICT,XX,YY,N2,M1,IN2,XW,YW,XVIEW,YVIEW,
1813.....1.....ZVIEW,0:0,0:0,0:,LHIDE,LBOX,0,1,N2,
1814.....2.....1,M1,0:05,1:0,0:2,0:95)
1815.....CALL SETCH(15,3,0,0,2)
1816.....WRITE(100,110) M,SIME,PEAK
1817.....DO 106 I=1,N2
1818.....DO 106 J=2,M1P1
1819...106 PICT(I,J-1)=F(1,I,J,2)/11600.
1820.....PEAK=-9.9E+99
1821.....DO 215 I=1,N2
1822.....DO 215 J=1,M1
1823.....IF(PICT(I,J).GT. PEAK) PEAK=PICT(I,J)
1824...215 CONTINUE
1825.....CALL FRAME
1826.....LHIDE=2
1827.....LBOX=3
1828.....CALL PICTURE(PICT,XX,YY,N2,M1,IN2,XW,YW,XVIEW,YVIEW,
1829.....1.....ZVIEW,0:0,0:0,0:,LHIDE,LBOX,0,1,N2,
1830.....2.....1,M1,0:05,1:0,0:2,0:95)
1831.....CALL SETCH(15,3,0,0,2)
1832.....WRITE(100,111) M,SIME,PEAK
1833...111 FORMAT("M...=",I4,3X,"TIME=",E9.2,"SEC:"/"TEMPERATURE",
1834.....12X,"(MAXIMUM=",E9.2,"EV:")")
1835...200 IF(IDNTP.NE.1) GO TO 300
1836...C....PLOT DENSITY & TEMP. AT Z=Z(JJZ1)
1837.....CALL FRAME
1838.....PEAK=-9.9E+99
1839.....DO 202 I=1,N2
1840.....C(I)=F(1,I,JJZ1,2)/11600.
1841.....IF(C(I).GT. PEAK) PEAK=C(I)
1842...202 CONTINUE
1843.....YMAX=5.0
1844.....IF(PEAK.GT. 5.AND. PEAK.LT.10.) YMAX=10.

```

```

1845.....IF (PEAK.GT.10.AND.PEAK.LT.20.) YMAX=20.
1846.....IF (PEAK.GT.20.AND.PEAK.LT.50.) YMAX=50.
1847.....IF (PEAK.GT.50.AND.PEAK.LT.100.) YMAX=100.
1848.....IF (PEAK.GT.100.) YMAX=PEAK+1.
1849.....CALL MAPS(X(1),X(N2),0.,YMAX,0.1,1.0,0.1,0.85)
1850.....CALL SETPCH(1,0,1,0.50)
1851.....CALL TRACEC(1H+,X,C,N2)
1852.....DO 204 I=1,N2
1853.....C(I)=W(1,I,JJZ1,3)
1854...204 CONTINUE
1855.....CALL TRACEC(1HD,X,C,N2)
1856.....Z1=RSMA+SZ(JJZ1)
1857.....CALL SETCH(10.,31.,0,0,2)
1858.....WRITE(100,205) M,SIME,Z1
1859...205 FORMAT("M...=",I4/"TIME=",E9.2," SEC. "/"D:DENSITY",2X,
1860.....1"+:TEMPERATURE (EV.)",2X,"AT-Z=",F6.2," CM.")
1861...300 IF (IFIEL.NE.1) GO TO 400
1862...C.....PLOT BZ & BPHI AT-Z=Z(JJZ2)
1863.....CALL FRAME
1864.....DO 304 I=1,N2
1865...304 C(I)=B3(I,JJZ2,1)/1000.
1866.....PEAK=AMAXAF(C,1,N2)
1867.....YMAX=10.
1868.....IF (PEAK.GT.10.) YMAX=15.
1869.....IF (PEAK.GT.15.) YMAX=20.
1870.....IF (PEAK.GT.20.) YMAX=30.
1871.....CALL MAPS(X(1),X(N2),-10.,YMAX,0.1,1.0,0.1,0.85)
1872.....CALL SETPCH(1,0,1,0.50)
1873.....CALL TRACEC(1HT,X,C,N2)
1874.....DO 302 I=1,N2
1875...302 C(I)=B2(I,JJZ2,1)/1000.
1876.....CALL TRACEC(1H2,X,C,N2)
1877.....CALL SETCH(10.,31.,0,0,2)
1878.....Z1=SZ(JJZ2)+RSMA
1879.....WRITE(100,305) M,SIME,Z1
1880...305 FORMAT("M...=",I4/"TIME=",E9.2," SEC. "/"Z:BZ FIELD",2X,
1881.....1"T:TOROIDAL FIELD (KG.)",2X,"AT-Z=",F6.2," CM.")
1882...400 CONTINUE
1883.....RETURN
1884.....END

```

```

1885
1886
1887
1888
1889
1890
1891 ..... SUBROUTINE CRDUT2(NL)
1892 C:..... THIS SUBROUTINE IS CALLED BY "ABIPSI"
1893 C:..... TO INVERT A TRI-DIAGONAL MATRIX.
1894 ..... COMMON /CRDUTV/ AA(IN2,3), B(IN2), X1(IN2)
1895 ..... OPTIMIZE
1896 ..... NN=NL-1
1897 ..... AA(1,1)=0.
1898 ..... AA(NL,3)=0.
1899 ..... AA(1,3)=AA(1,3)/AA(1,2)
1900 ..... B(1)=B(1)/AA(1,2)
1901 ..... DO 1 II=1,NN
1902 ..... I=II+1
1903 ..... AA(I,2)=AA(I,2)-AA(I,1)*AA(I-1,3)
1904 ..... AA(I,3)=AA(I,3)/AA(I,2)
1905 ..... B(I)=(B(I)-AA(I,1)*B(I-1))/AA(I,2)
1906 ..... 1 CONTINUE
1907 C:..... BACKWARD SUBSTITUTION
1908 ..... X1(NL)=B(NL)
1909 ..... DO 2 II=1,NN
1910 ..... I=NL-II
1911 ..... X1(I)=B(I)-AA(I,3)*X1(I+1)
1912 ..... 2 CONTINUE
1913 ..... RETURN
1914 ..... END
1915
1916
1917
1918
1919
1920
1921 ..... SUBROUTINE COEF(I,J,A,B,C,D,E)
1922 C:..... THIS SUBROUTINE IS CALLED BY "CGCOEF" TO CALCULATE
1923 C:..... THE COEFFICIENTS OF THE FIVE-POINT DIFFERENCE EQUATION
1924 C:..... FOR THE VACUUM EQUATION FOR PSI.

```

```

1925.....COMMON /MESH/ RR (NT), ZZ (MT), DELRR (NT-1), DELZZ (MT-1),
1926.....1 SR (IN2), SZ (JM1P1), DELR (IN2-1), DELZ (JM1P1-1)
1927.....COMMON /CONST/ N1, N2, N3, M1, M2, N1P1, N1P2, N2P1, N2P2, N2M1,
1928.....1N1PN2, N1PN2P, N1PN2M, M1P1, M1M1, M1P2, M2P1, M2M1, M2P2, NN, NNM1, MM, MMM1
1929.....COMMON /DIM/ RSMA, RB, BB1, RLNGTH, ZLNGTH
1930.....RRDEL = (DELRR (I) + DELRR (I-1)) / 2.
1931.....ZZDEL = (DELZZ (J) + DELZZ (J-1)) / 2.
1932.....ALPHA = RR (I) * (1.0 - RLNGTH * RR (I))
1933.....RRP12 = (RR (I) + RR (I+1)) / 2.
1934.....RRM12 = (RR (I) + RR (I-1)) / 2.
1935.....BETAP2 = RRP12 * 3 / (1. - RLNGTH * RRP12)
1936.....BETAM2 = RRM12 * 3 / (1. - RLNGTH * RRM12)
1937.....ZETAP2 = (ZZ (J) + ZZ (J+1)) / 2.
1938.....ZETAM2 = (ZZ (J) + ZZ (J-1)) / 2.
1939.....A = -ALPHA * BETAM2 / (RRDEL * DELRR (I-1))
1940.....B = - (ZZ (J) * 2) * (ZETAM2 * 2) / (ZZDEL * DELZZ (J-1))
1941.....C = ALPHA / RRDEL * (BETAP2 / DELRR (I) + BETAM2 / DELRR (I-1))
1942.....1 + ZZ (J) * 2 / ZZDEL * (ZETAP2 * 2 / DELZZ (J) + ZETAM2 * 2 / DELZZ (J-1))
1943.....D = -ALPHA * BETAP2 / (RRDEL * DELRR (I))
1944.....E = - (ZZ (J) * 2) * (ZETAP2 * 2) / (ZZDEL * DELZZ (J))
1945.....IF (J.NE.MMM1) GO TO 10
1946.....B = B + E
1947.....E = 0.
1948.....10 CONTINUE
1949.....RETURN
1950.....END
1951
1952
1953
1954
1955
1956
1957.....SUBROUTINE ADIPSI (M)
1958 C..... THIS SUBROUTINE IS CALLED BY "FLX" TO CALCULATE
1959 C..... PSI IN THE PLASMA REGION USING AN ADI SCHEME.
1960.....LCM (VARIABLES), (LCMICCG), (CDEFAD), (CONTOUR), (WLCM2)
1961.....COMMON /VARIABLES/ PSI (IN2, JM1P1, 3), B1 (IN2, JM1P1, 2),
1962.....1B2 (IN2, JM1P1, 2), B3 (IN2, JM1P1, 2), X1 (IN2, JM1P1, 2), C1 (IN2, JM1P1, 2),
1963.....1C2 (IN2, JM1P1, 2), C3 (IN2, JM1P1, 2),
1964.....2T (IN2, JM1P1, 2), W (4, IN2, JM1P1, 3), F (4, IN2, JM1P1, 3),

```

```

1965.....3EI(IN2,JM1P1,2),CN(IN2,JM1P1,2)
1966.....COMMON/CONST2/IS,IB,IE,ID,IP,JS,JB,JE,JD,JP,BX,BY,BATA,BATB
1967.....COMMON/MESH/RR(NT),ZZ(MT),BELRR(NT-1),BELZZ(MT-1),
1968.....1SR(IN2),SZ(JM1P1),BELR(IN2-1),BELZ(JM1P1-1)
1969.....COMMON/GRDUTY/AA(IN2,3),B(IN2),X1(IN2)
1970.....COMMON/CONST/N1,N2,N3,M1,M2,N1P1,N1P2,N2P1,N2P2,N2M1,
1971.....1N1PN2,N1PN2P,N1PN2M,M1P1,M1M1,M1P2,M2P1,M2M1,M2P2,NN,NM1,MM,MMM1
1972.....COMMON/CUR/NCUR,LOCUR(35),VALCUR(35)
1973.....COMMON/DIM/RSMA,RB,BB1,RLNGTH,ZLNGTH
1974.....COMMON/COEFAD/COEFR1(IN2-1,JM1P1-1),COEFR2(IN2-1,JM1P1-1),
1975.....2COEFR3(IN2-1,JM1P1-1),COEFZ1(IN2-1,JM1P1-1),
1976.....3COEFZ2(IN2-1,JM1P1-1),COEFZ3(IN2-1,JM1P1-1)
1977.....COMMON/CONTOUR/FLUXRZ(NT,MT),RAXIS(NT),ZAXIS(MT)
1978.....COMMON/VAR2/FR,FK,CV,C,Q,HU,UN,BT,TMAX,PTIME1,PTIME2,TIME,
1979.....1.TDIS1,TDIS2,
1980.....1BMIN,SGO,CDE,MW,MP,MC,NEURY,NCHECK,BTOR,BN,TEMP,
1981.....2CUR1MAX,CUR2MAX,CUR3MAX,CURTOR,RSPINCH,RBPINCH
1982.....COMMON/WSCM/S(N,(-4,4))
1983.....COMMON/WLCM2/GA(IN2,JM1P1),GB(IN2,JM1P1),GC(IN2,JM1P1),
1984.....1GB(IN2,JM1P1),GE(IN2,JM1P1),GF(IN2,JM1P1),WSCM1(4,IN2,JM1P1),
1985.....2WSCM2(4,IN2,JM1P1),APSI(NT,MT)
1986.....COMMON/LEMICG/MTRX(N,(-4,4)),LU(N,(-4,4)),X(N),Y(N)
1987.....REAL MTRX,LU
1988.....OPTIMIZE
1989.....CALL QACOPY(WSCM1,W(1,1,1,1),4*NS)
1990.....CALL QACOPY(WSCM2,W(1,1,1,2),4*NS)
1991.....CALL QACOPY(GA,PSI(1,1,1),NS)
1992.....CALL QACOPY(GB,PSI(1,1,2),NS)
1993.....IF(MOD(M,2).EQ.1)GO TO 10
1994 C.....IMPLICIT IN R-DIRECTION
1995.....DO 1 J=2,M1
1996.....DO 2 I=2,N2M1
1997.....II=I-1
1998.....AA(II,1)=-COEFR1(II,J-1)
1999.....AA(II,2)=+COEFR2(II,J-1)
2000.....AA(II,3)=-COEFR3(II,J-1)
2001.....B(II)=GA(I,J)+CDE/(T(I,J,1)+BY+BY)
2002.....1+(GA(I,J+1)+GA(I,J-1)-2*(GA(I,J)))
2003.....3-BT+WSCM1(3,I,J)/WSCM1(1,I,J)/(2*(BY)+(GA(I,J+1)-GA(I,J-1)))
2004.....2-CONTINUE
2005 C.....TAKE CARE OF THE BOUNDARY CONDITIONS:

```

```

2006.....I=2
2007.....II=I-1
2008.....B(II)=B(II)-AA(II,1)*PSI(I-1,J,3)
2009.....I=N2M1
2010.....II=I-1
2011.....B(II)=B(II)-AA(II,3)*PSI(I+1,J,3)
2012.....CALL CRDUT2(N2-2)
2013.....DO 5 I=2,N2M1
2014.....5 PSI(I,J,3)=X1(I-1)
2015.....1 CONTINUE
2016.....DO 6 I=2,N2M1
2017.....6 PSI(I,1,3)=PSI(I,3,3)
2018.....RETURN
2019 C.....IMPLICIT IN Z-DIRECTION
2020.....10 CONTINUE
2021.....DO 11 I=2,N2M1
2022.....II=I-1
2023.....DO 12 J=2,M1
2024.....JJ=J-1
2025.....AA(JJ,1)=-CDEFZ1(II,JJ)
2026.....AA(JJ,2)=+CDEFZ2(II,JJ)
2027.....AA(JJ,3)=-CDEFZ3(II,JJ)
2028.....B(JJ)=GA(I,J)+CDE*SR(I)/(T(I,J,1)*BX*BX)
2029.....1*((GA(I+1,J)-GA(I,J))/(SR(I+1)+SR(I))/2.)
2030.....2-(GA(I,J)-GA(I-1,J))/(SR(I-1)+SR(I))/2.)
2031.....4-BT*VSEM1(2,I,J)/VSEM1(1,I,J)*(GA(I+1,J)-GA(I-1,J))/(2.*BX)
2032.....12 CONTINUE
2033 C.....DO B,C: S
2034.....AA(1,3)=AA(1,3)+AA(1,1)
2035.....J=M1
2036.....B(J-1)=B(J-1)-AA(J-1,3)*PSI(I,J+1,3)
2037.....CALL CRDUT2(M1-1)
2038.....DO 15 J=2,M1
2039.....15 PSI(I,J,3)=X1(J-1)
2040.....PSI(I,1,3)=PSI(I,3,3)
2041.....11 CONTINUE
2042.....RETURN
2043.....END
2044

```

```

2045
2046.....SUBROUTINE XIMCR0
2047 C..... THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM
2048 C..... TO CALCULATE MASS & MOMENTUM DENSITIES; THIS IS THE
2049 C..... IMPLICIT IN-R STEP.
2050..... LCM (VARIABLES), (LCMICCG), (CDEFAD), (CONTOUR), (WLCM2)
2051..... COMMON/VARIABLES/PSI (IN2, JM1P1, 3), B1 (IN2, JM1P1, 2),
2052..... 1B2 (IN2, JM1P1, 2), B3 (IN2, JM1P1, 2), XI (IN2, JM1P1, 2), C1 (IN2, JM1P1, 2),
2053..... 1C2 (IN2, JM1P1, 2), C3 (IN2, JM1P1, 2),
2054..... 2T (IN2, JM1P1, 2), W (4, IN2, JM1P1, 3), F (4, IN2, JM1P1, 3),
2055..... 3EI (IN2, JM1P1, 2), CN (IN2, JM1P1, 2)
2056..... COMMON/CONST2/IS, IB, IE, ID, IP, JS, JB, JE, JD, JP, BX, BY, BATA, BATB
2057..... COMMON /MESH/ RR (NT), ZZ (MT), DELRR (NT-1), DELZZ (MT-1),
2058..... 1SR (IN2), SZ (JM1P1), DELR (IN2-1), DELZ (JM1P1-1)
2059..... COMMON/CONST/ N1, N2, N3, M1, M2, N1P1, N1P2, N2P1, N2P2, N2M1,
2060..... 1N1PN2, N1PN2P, N1PN2M, M1P1, M1M1, M1P2, M2P1, M2M1, M2P2, NN, NNM1, MM, MMM1
2061..... COMMON/CUR/ NCUR, LCCUR (35), VALCUR (35)
2062..... COMMON/DIM/RMA, RB, BB1, RLNGTH, ZLNGTH
2063..... COMMON/CDEFAD/CDEFR1 (IN2-1, JM1P1-1), CDEFR2 (IN2-1, JM1P1-1),
2064..... 2CDEFR3 (IN2-1, JM1P1-1), CDEFZ1 (IN2-1, JM1P1-1),
2065..... 3CDEFZ2 (IN2-1, JM1P1-1), CDEFZ3 (IN2-1, JM1P1-1)
2066..... COMMON/CONTOUR/ FLUXR2 (NT, MT), RAXIS (NT), ZAXIS (MT)
2067..... COMMON/VAR2/FR, FK, CV, C, Q, HU, UN, BT, TMAX, PTIME1, PTIME2, TIME,
2068..... 1 TDIS1, TDIS2,
2069..... 1BMIN, SGO, CDE, MW, MP, MC, NCURV, NCHECK, BTOR, DN, TEMP,
2070..... 2CUR1MAX, CUR2MAX, CUR3MAX, CURTOR, RSPINCH, RBPINCH
2071..... COMMON /WSEM/ S (N, (-4, 4))
2072..... COMMON/WLCM2/ GA (IN2, JM1P1), GB (IN2, JM1P1), GC (IN2, JM1P1),
2073..... 1GD (IN2, JM1P1), GE (IN2, JM1P1), GF (IN2, JM1P1), VSEM1 (4, IN2, JM1P1),
2074..... 2VSEM2 (4, IN2, JM1P1), APSI (NT, MT)
2075..... COMMON /LCMICCG/ MTRX (N, (-4, 4)), LU (N, (-4, 4)), X (N), Y (N)
2076..... REAL MTRX, LU
2077..... DIMENSION AA (4*(IN2-1), 12), BB (4, 4)
2078..... OPTIMIZE
2079..... TMALL=1.0E-30
2080..... K=1
2081..... K2=3
2082..... SW=1.0
2083..... IF (TIME.GT.TDIS1) SW=1.0-2.0*BT/(BX+BX+TDIS2)
2084..... CALL QACOPY (VSEM1, W (1, 1, 1, 1), 4*NS)

```

```

2085.....CALL QACOPY(VSEM2,W(1,1,1,2),4*NS)
2086.....CALL QACOPY(GA,EI(1,1,1),NS)
2087.....CALL QACOPY(GB,EI(1,1,2),NS)
2088.....DO 41 J=JB,JE
2089.....DO 42 I=IB,IE
2090.....HON=0.4
2091.....IF (VSEM1(1,I,J).LT.0.3) HON=2.0
2092.....C.....ADD MORE DISS. TO THE UPPER RIGHT HAND CORNER.
2093.....C.....IF (TIME.GT.TBIS1.AND.I.GE.5.AND.J.GE.8) HON=2.0+HON
2094.....888 F(1,I,J,K2)=VSEM1(1,I,J)
2095.....2-BATB*(VSEM1(3,I,J+1)-VSEM1(3,I,J-1))
2096.....3+(F(1,I,J,K)+F(1,I,J,K+1))
2097.....6+HON/4.*0.5*BT/BX*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)
2098.....7/VSEM1(1,I,J))*VSEM1(1,I+1,J)-VSEM1(1,I,J))-ABS(VSEM1(2,I,J)/VSEM
2099.....91(1,I,J)
2100.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))*VSEM1(1,I,J)-VSEM1(1,I-1,J))
2101.....6+HON/4.*0.5*BT/BY*(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
2102.....7/VSEM1(1,I,J))*VSEM1(1,I,J+1)-VSEM1(1,I,J))-ABS(VSEM1(3,I,J)/VSEM
2103.....91(1,I,J)
2104.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))*VSEM1(1,I,J)-VSEM1(1,I,J-1))
2105.....F(2,I,J,K2)=SW*VSEM1(2,I,J)+0.5*(1.0-SW)*(VSEM1(2,I-1,J)+VSEM1(2,I
2106.....9+1,J))
2107.....4-BATB*(VSEM1(2,I,J+1)+VSEM1(3,I,J+1)
2108.....5/VSEM1(1,I,J+1)-VSEM1(2,I,J-1)+VSEM1(3,I,J-1)/VSEM1(1,I,J-1))
2109.....6+(F(2,I,J,K)+F(2,I,J,K+1))
2110.....6+HON/4.*0.5*BT/BX*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)
2111.....7/VSEM1(1,I,J))*VSEM1(2,I+1,J)-VSEM1(2,I,J))-ABS(VSEM1(2,I,J)/VSEM
2112.....91(1,I,J)
2113.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))*VSEM1(2,I,J)-VSEM1(2,I-1,J))
2114.....6+HON/4.*0.5*BT/BY*(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
2115.....7/VSEM1(1,I,J))*VSEM1(2,I,J+1)-VSEM1(2,I,J))-ABS(VSEM1(3,I,J)/VSEM
2116.....91(1,I,J)
2117.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))*VSEM1(2,I,J)-VSEM1(2,I,J-1))
2118.....F(3,I,J,K2)=SW*VSEM1(3,I,J)+0.5*(1.0-SW)*(VSEM1(3,I-1,J)+VSEM1(3,I
2119.....9+1,J))
2120.....3-BATB*((VSEM1(3,I,J+1)+2/VSEM1(1,I,J+1)+
2121.....4(FK-1.)*VSEM1(1,I,J+1)+EI(1,J+1,K))-VSEM1(3,I,J-1)+2/
2122.....5VSEM1(1,I,J-1)+(FK-1.)*VSEM1(1,I,J-1)+GA(1,J-1))
2123.....6+(F(3,I,J,K)+F(3,I,J,K+1))
2124.....6+HON/4.*0.5*BT/BX*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)

```



```

2125.....7/VSEM1(1,I,J) + (VSEM1(3,I+1,J) - VSEM1(3,I,J)) - ABS(VSEM1(2,I,J)/VSEM
2126.....91(1,I,J)
2127.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J) + (VSEM1(3,I,J) - VSEM1(3,I-1,J))
2128.....6+HBN/4. + 0.5*BT/BY + (ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1) - VSEM1(3,I,J)
2129.....7/VSEM1(1,I,J) + (VSEM1(3,I,J+1) - VSEM1(3,I,J)) - ABS(VSEM1(3,I,J)/VSEM
2130.....91(1,I,J)
2131.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1) + (VSEM1(3,I,J) - VSEM1(3,I,J-1))
2132.....F(4,I,J,K2) = SW + VSEM1(4,I,J) + 0.5*(1.0-SW) + (VSEM1(4,I-1,J) + VSEM1(4,I
2133.....9+1,J))
2134.....3-BATB + (VSEM1(3,I,J+1) + VSEM1(4,I,J+1)
2135.....4/VSEM1(1,I,J+1) - VSEM1(3,I,J-1) + VSEM1(4,I,J-1)/VSEM1(1,I,J-1))
2136.....5+(F(4,I,J,K) + F(4,I,J,K+1))
2137.....6+HBN/4. + 0.5*BT/BX + (ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J) - VSEM1(2,I,J)
2138.....7/VSEM1(1,I,J) + (VSEM1(4,I+1,J) - VSEM1(4,I,J)) - ABS(VSEM1(2,I,J)/VSEM
2139.....91(1,I,J)
2140.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J) + (VSEM1(4,I,J) - VSEM1(4,I-1,J))
2141.....6+HBN/4. + 0.5*BT/BY + (ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1) - VSEM1(3,I,J)
2142.....7/VSEM1(1,I,J) + (VSEM1(4,I,J+1) - VSEM1(4,I,J)) - ABS(VSEM1(3,I,J)/VSEM
2143.....91(1,I,J)
2144.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1) + (VSEM1(4,I,J) - VSEM1(4,I,J-1))
2145.....42 CONTINUE
2146.....41 CONTINUE
2147.....DD 100 I=IB,IE
2148.....100 F(3,I,JE,K2) = F(3,I,JE,K2) + 2.0*BATB + VSEM1(3,I,JE+1) + 2/VSEM1(1,I,JE
2149.....1+1)
2150.....DD 153 J=JB,JE
2151.....F(2,2,J,K+2) = F(2,2,J,K+2) + (FK-1.) + VSEM2(1,IS,J) + 6B(IS,J)
2152.....1 + BATA + SR(IS)/SR(2)
2153.....F(2,IE,J,K+2) = F(2,IE,J,K+2) - (FK-1.) + VSEM2(1,IB,J) + 6B(IB,J)
2154.....1 + BATA + SR(IE)/SR(IE)
2155.....F(1,IB,J,K+2) = F(1,IB,J,K+2) + BATA + SR(IE+1)/SR(IE) +
2156.....1VSEM2(2,IB-1,J)
2157.....F(1,IE,J,K+2) = F(1,IE,J,K+2) - BATA + SR(IE+1)/SR(IE) +
2158.....1VSEM2(2,IE+1,J)
2159.....F(2,IB,J,K+2) = F(2,IB,J,K+2) - BATA + SR(IE+1)/SR(IE) +
2160.....1VSEM2(2,IB-1,J) + 2/VSEM2(1,IB-1,J)
2161.....F(2,IE,J,K+2) = F(2,IE,J,K+2) + BATA + SR(IE+1)/SR(IE) +
2162.....1VSEM2(2,IE+1,J) + 2/VSEM2(1,IE+1,J)
2163.....F(3,IB,J,K+2) = F(3,IB,J,K+2) + BATA + SR(IE+1)/SR(IE) +
2164.....1VSEM2(2,IB-1,J) + VSEM2(3,IB-1,J)/VSEM2(1,IB-1,J)
2165.....F(3,IE,J,K+2) = F(3,IE,J,K+2) - BATA + SR(IE+1)/SR(IE) +

```

```

2166.....1VSEM2(2,IE+1,J)+VSEM2(3,IE+1,J)/VSEM2(1,IE+1,J)
2167.....F(4,IB,J,K+2)=F(4,IB,J,K+2)+BATA*SR(IB-1)/SR(IB)+
2168.....1VSEM2(2,IB-1,J)+VSEM2(4,IB-1,J)/VSEM2(1,IB-1,J)
2169.....F(4,IE,J,K+2)=F(4,IE,J,K+2)-BATA*SR(IE+1)/SR(IE)+
2170.....1VSEM2(2,IE+1,J)+VSEM2(4,IE+1,J)/VSEM2(1,IE+1,J)
2171.....DO 154 I=IS,IP
2172.....IF (I.EQ. IS) GO TO 155
2173.....AA(4*I-3,1)=0:
2174.....AA(4*I-3,2)=-BATA*SR(I)/SR(I+1)
2175.....AA(4*I-3,3)=0:
2176.....AA(4*I-3,4)=0:
2177.....AA(4*I-2,1)=-BATA*SR(I)/SR(I+1)+((FK-1.)+GB(I,J))
2178.....AA(4*I-2,2)=-BATA*SR(I)/SR(I+1)+VSEM2(2,I,J)/
2179.....1.....VSEM2(1,I,J)
2180.....AA(4*I-2,3)=0:
2181.....AA(4*I-2,4)=0:
2182.....AA(4*I-1,1)=0:
2183.....AA(4*I-1,2)=0:
2184.....AA(4*I-1,3)=-BATA*SR(I)/SR(I+1)+VSEM2(2,I,J)/VSEM2(1,I,J)
2185.....AA(4*I-1,4)=0:
2186.....AA(4*I...,1)=0:
2187.....AA(4*I...,2)=0:
2188.....AA(4*I...,3)=0:
2189.....AA(4*I...,4)=-BATA*SR(I)/SR(I+1)+VSEM2(2,I,J)/VSEM2(1,I,J)
2190...155 CONTINUE
2191.....IF (I.EQ. IP) GO TO 156
2192.....AA(4*I-3,9)=0:
2193.....AA(4*I-3,10)=+BATA*SR(I+2)/SR(I+1)
2194.....AA(4*I-3,11)=0:
2195.....AA(4*I-3,12)=0:
2196.....AA(4*I-2,9)=+BATA*SR(I+2)/SR(I+1)+((FK-1.)+GB(I+2,J))
2197.....AA(4*I-2,10)=BATA*SR(I+2)/SR(I+1)+VSEM2(2,I+2,J)/
2198.....1.....VSEM2(1,I+2,J)
2199.....AA(4*I-2,11)=0:
2200.....AA(4*I-2,12)=0:
2201.....AA(4*I-1,9)=0:
2202.....AA(4*I-1,10)=0:
2203.....AA(4*I-1,11)=+BATA*SR(I+2)/SR(I+1)+VSEM2(2,I+2,J)/VSEM2(1,I+2,J)
2204.....AA(4*I-1,12)=0:
2205.....AA(4*I...,9)=0:

```

```

2206.....AA(4+I,10)=0:
2207.....AA(4+I,11)=0:
2208.....AA(4+I,12)=+BATA*SR(I+2)/SR(I+1)+WSEM2(2,I+2,J)/WSEM2(1,I+2,J)
2209...156 CONTINUE
2210.....AA(4+I-3,5)=1:
2211.....AA(4+I-3,6)=0:
2212.....AA(4+I-3,7)=0:
2213.....AA(4+I-3,8)=0:
2214.....AA(4+I-2,5)=-BT*(FK-1.0)+GB(I+1,J)/(2.0*SR(I+1))
2215.....AA(4+I-2,6)=1:
2216.....AA(4+I-2,7)=0:
2217.....AA(4+I-2,8)=-BT/SR(I+1)+WSEM2(4,I+1,J)/WSEM2(1,I+1,J)/2.0
2218.....AA(4+I-1,5)=0:
2219.....AA(4+I-1,6)=0:
2220.....AA(4+I-1,7)=1:
2221.....AA(4+I-1,8)=0:
2222.....AA(4+I,5)=0:
2223.....AA(4+I,6)=0:
2224.....AA(4+I,7)=0:
2225.....AA(4+I,8)=1.0+BT/(2.0*SR(I+1))+WSEM2(2,I+1,J)/WSEM2(1,I+1,J)
2226...154 CONTINUE
2227 C.....MODIFICATIONS.....:
2228 C
2229.....DO 499 LI=1,4
2230.....DO 499 LJ=5,12
2231.....IF (ABS (AA(LI,LJ)) .LT. TMAX) AA(LI,LJ)=0:
2232...499 CONTINUE
2233 C
2234.....DO 498 LJ=6,12
2235...498 AA(1,LJ)=AA(1,LJ)/AA(1,5)
2236 C
2237.....DO 500 LI=2,4
2238...500 AA(LI,6)=AA(LI,6)-AA(LI,5)+AA(1,6)
2239 C
2240.....DO 501 LI=5,8
2241...501 AA(LI,2)=AA(LI,2)-AA(LI,1)+AA(1,6)
2242 C
2243.....DO 502 LJ=7,12
2244...502 AA(2,LJ)=(AA(2,LJ)-AA(2,5)+AA(1,LJ))/AA(2,6)
2245 C

```

```

2246.....DD·503·LI=3,4
2247.....DD·504·LK=1,2
2248...504·AA(LI,7)=AA(LI,7)-AA(LI,4+LK)♦AA(LK,7)
2249...503·CONTINUE
2250·C
2251.....DD·505·LI=5,8
2252.....DD·506·LK=1,2
2253...506·AA(LI,3)=AA(LI,3)-AA(LI,LK)♦AA(LK,7)
2254...505·CONTINUE
2255·C
2256.....DD·507·LJ=8,12
2257.....DD·508·LK=1,2
2258...508·AA(3,LJ)=AA(3,LJ)-AA(3,4+LK)♦AA(LK,LJ)
2259.....AA(3,LJ)=AA(3,LJ)/AA(3,7)
2260...507·CONTINUE
2261·C
2262.....DD·509·LK=1,3
2263...509·AA(4,8)=AA(4,8)-AA(4,4+LK)♦AA(LK,8)
2264·C
2265.....DD·510·LI=5,8
2266.....DD·511·LK=1,3
2267...511·AA(LI,4)=AA(LI,4)-AA(LI,LK)♦AA(LK,8)
2268...510·CONTINUE
2269·C
2270.....DD·512·LJ=9,12
2271.....DD·513·LK=1,3
2272...513·AA(4,LJ)=AA(4,LJ)-AA(4,4+LK)♦AA(LK,LJ)
2273.....AA(4,LJ)=AA(4,LJ)/AA(4,8)
2274...512·CONTINUE
2275·C
2276.....F(1,IB,J,K+2)=F(1,IB,J,K+2)/AA(1,5)
2277.....DD·520·L=2,4
2278.....LE=L-1
2279.....DD·521·LK=1,LE
2280.....IF (ABS (AA (L,4+LK)) .LT. TMAXN) ·AA (L,4+LK) =0.
2281.....IF (ABS (F (LK,IB,J,K+2)) .LT. TMAXL) ·F (LK,IB,J,K+2) =0.
2282...521·F(L,IB,J,K+2)=F(L,IB,J,K+2)
2283...1·-AA(L,4+LK)♦F(LK,IB,J,K+2)
2284.....F(L,IB,J,K+2)=F(L,IB,J,K+2)/AA(L,4+L)
2285...520·CONTINUE

```

```

2286·C
2287·C·....END·OF·MODIFICATIONS....
2288·.....IPP=4+IP
2289·.....DO·165·I=5,IPP,4
2290·.....DO·166·L=1,4
2291·.....DO·167·LL=1,4
2292·.....BB(L,LL)=0.
2293·.....DO·168·LLL=1,4
2294·.....IF·(ABS(AA(I+L-1,LLL))·.LT·TMALL)·AA(I+L-1,LLL)=0.
2295·.....IF·(ABS(AA(I+LLL-5,8+LL))·.LT·TMALL)·AA(I+LLL-5,8+LL)=0.
2296·...168·BB(L,LL)=BB(L,LL)+AA(I+L-1,LLL)+AA(I+LLL-5,8+LL)
2297·...167·CONTINUE
2298·...166·CONTINUE
2299·.....DO·169·L=1,4
2300·...169·AA(I-1+L,5)=AA(I-1+L,5)-BB(L,1)
2301·.....DO·170·L=1,3
2302·...170·AA(I,5+L)=(AA(I,5+L)-BB(1,L+1))/AA(I,5)
2303·.....IF·(I·EQ·(IPP-3))·GO·TO·971
2304·.....DO·171·L=1,4
2305·...171·AA(I,8+L)=AA(I,8+L)/AA(I,5)
2306·...971·CONTINUE
2307·.....DO·172·L=1,3
2308·...172·AA(I+L,6)=AA(I+L,6)-BB(L+1,2)-AA(I+L,5)+AA(I,6)
2309·.....DO·173·L=1,2
2310·...173·AA(I+1,6+L)=(AA(I+1,6+L)-BB(2,2+L)-AA(I+1,5)+AA(I,6+L))/AA(I+1,6)
2311·.....IF·(I·EQ·(IPP-3))·GO·TO·191
2312·.....DO·174·L=1,4
2313·.....AA(I+L+3,2)=AA(I+L+3,2)-AA(I+L+3,1)+AA(I,6)
2314·.....AA(I+1,8+L)=(AA(I+1,8+L)-AA(I+1,5)+AA(I,8+L))/AA(I+1,6)
2315·...174·CONTINUE
2316·...191·CONTINUE
2317·.....DO·175·L=1,2
2318·...175·AA(I+L+1,7)=AA(I+L+1,7)-BB(2+L,3)-AA(I+L+1,5)+AA(I,7)-
2319·...1AA(I+L+1,6)+AA(I+1,7)
2320·.....IF·(ABS(AA(I+2,5))·.LT·TMALL)·AA(I+2,5)=0.
2321·.....IF·(ABS(AA(I+1,8))·.LT·TMALL)·AA(I+1,8)=0.
2322·.....IF·(ABS(AA(I+2,6))·.LT·TMALL)·AA(I+2,6)=0.
2323·.....AA(I+2,8)=(AA(I+2,8)-BB(3,4)-AA(I+2,5)+AA(I,8)-AA(I+2,6)+AA(I+1,8)
2324·.....2)/AA(I+2,7)
2325·.....IF·(I·EQ·(IPP-3))·GO·TO·190

```

```

2326.....DD·176·L=1,4
2327.....DD·177·LL=1,2
2328.....AA(I+L+3,3)=AA(I+L+3,3)-AA(I+L+3,LL)♦AA(I+LL-1,7)
2329.....AA(I+2,8+L)=AA(I+2,8+L)-AA(I+2,4+LL)♦AA(I+LL-1,8+L)
2330.....177·CONTINUE
2331.....AA(I+2,8+L)=AA(I+2,8+L)/AA(I+2,7)
2332.....176·CONTINUE
2333.....190·CONTINUE
2334.....IF(ABS(AA(I+3,7)):LT:TMALL)·AA(I+3,7)=0:
2335.....AA(I+3,8)=AA(I+3,8)-BB(4,4)-AA(I+3,5)♦AA(I,8)-AA(I+3,6)♦AA(I+1,8)
2336.....1-AA(I+3,7)♦AA(I+2,8)
2337.....IF(I·EQ·(IPP-3))·GO·TO·180
2338.....DD·178·L=1,4
2339.....DD·179·LL=1,3
2340.....IF(ABS(AA(I+L+3,LL)):LT:TMALL)·AA(I+L+3,LL)=0:
2341.....IF(ABS(AA(I+LL-1,8)):LT:TMALL)·AA(I+LL-1,8)=0:
2342.....AA(I+L+3,4)=AA(I+L+3,4)-AA(I+L+3,LL)♦AA(I+LL-1,8)
2343.....AA(I+3,8+L)=AA(I+3,8+L)-AA(I+3,4+LL)♦AA(I+LL-1,8+L)
2344.....179·CONTINUE
2345.....AA(I+3,8+L)=AA(I+3,8+L)/AA(I+3,8)
2346.....178·CONTINUE
2347.....180·CONTINUE
2348.....DD·181·L=1,4
2349.....DD·182·LL=1,4
2350.....IF(ABS(AA(I+L-1,LL)):LT:TMALL)·AA(I+L-1,LL)=0:
2351.....IF(ABS(F(LL,(I+7)/4-1,J,K+2)):LT:TMALL)
2352.....1F(LL,(I+7)/4-1,J,K+2)=0:
2353.....182·F(L,(I+7)/4,J,K+2)=F(L,(I+7)/4,J,K+2)-AA(I+L-1,LL)♦F(LL,(I+7)/4-1
2354.....1,J,K+2)
2355.....LE=L-1
2356.....IF(L·EQ·1)·GO·TO·184
2357.....DD·183·LL=1,LE
2358.....F(L,(I+7)/4,J,K+2)=F(L,(I+7)/4,J,K+2)-AA(I+L-1,4+LL)♦F(LL,(I+7)/4
2359.....1,J,K+2)
2360.....183·CONTINUE
2361.....184·CONTINUE
2362.....F(L,(I+7)/4,J,K+2)=F(L,(I+7)/4,J,K+2)/AA(I+L-1,5+L-1)
2363.....181·CONTINUE
2364.....165·CONTINUE
2365.....W(4,IE,J,K+2)=F(4,IE,J,K+2)
2366.....W(3,IE,J,K+2)=F(3,IE,J,K+2)-AA(IPP-1,8)♦W(4,IE,J,K+2)

```

```

2367.....W(2,IE,J,K+2)=F(2,IE,J,K+2)-AA(IPP-2,8)*W(4,IE,J,K+2)
2368.....1-AA(IPP-2,7)*W(3,IE,J,K+2)
2369.....W(1,IE,J,K+2)=F(1,IE,J,K+2)-AA(IPP-3,8)*W(4,IE,J,K+2)
2370.....1-AA(IPP-3,7)*W(3,IE,J,K+2)-AA(IPP-3,6)*W(2,IE,J,K+2)
2371.....BO=192-I=5,IPP,4
2372.....II=IPP+5-I-4
2373.....BO=193-L=1,4
2374.....BB(1,L)=0.
2375.....BO=194-LL=1,4
2376.....IF.(ABS(AA(II+1-L,8+LL)).LT.TMALL).AA(II+1-L,8+LL)=0.
2377.....IF.(ABS(W(LL,(II+4)/4+1,J,K+2)).LT.TMALL)
2378.....1W(LL,(II+4)/4+1,J,K+2)=0.
2379.....194-BB(1,L)=BB(1,L)+AA(II+1-L,8+LL)*W(LL,(II+4)/4+1,J,K+2)
2380.....193-CONTINUE
2381.....W(4,(II+4)/4,J,K+2)=F(4,(II+4)/4,J,K+2)-BB(1,1)
2382.....W(3,(II+4)/4,J,K+2)=F(3,(II+4)/4,J,K+2)-BB(1,2)
2383.....1-AA(II-1,8)*W(4,(II+4)/4,J,K+2)
2384.....W(2,(II+4)/4,J,K+2)=F(2,(II+4)/4,J,K+2)-BB(1,3)
2385.....1-AA(II-2,8)*W(4,(II+4)/4,J,K+2)-AA(II-2,7)*W(3,(II+4)/4,J,K+2)
2386.....W(1,(II+4)/4,J,K+2)=F(1,(II+4)/4,J,K+2)-BB(1,4)
2387.....1-AA(II-3,8)*W(4,(II+4)/4,J,K+2)-AA(II-3,7)*W(3,(II+4)/4,J,K+2)
2388.....2-AA(II-3,6)*W(2,(II+4)/4,J,K+2)
2389.....192-CONTINUE
2390.....153-CONTINUE
2391.....RETURN
2392.....END
2393
2394
2395.....SUBROUTINE YIMCRD
2396.....C..... SAME AS XIMCRD, EXCEPT THIS IS THE IMPLICIT IN-2 STEP.
2397.....LCM(VARIABLES), (LCMICCG), (CDEFAB), (CONTOUR), (WLCM2)
2398.....COMMON/VARIABLES/PSI(IN2,JM1P1,3),B1(IN2,JM1P1,2),
2399.....1B2(IN2,JM1P1,2),B3(IN2,JM1P1,2),XI(IN2,JM1P1,2),C1(IN2,JM1P1,2),
2400.....1C2(IN2,JM1P1,2),C3(IN2,JM1P1,2),
2401.....2T(IN2,JM1P1,2),W(4,IN2,JM1P1,3),F(4,IN2,JM1P1,3),
2402.....3EI(IN2,JM1P1,2),CN(IN2,JM1P1,2)
2403.....COMMON/CONST2/IS,IB,IE,ID,IP,JS,JB,JE,JB,JP,BX,BY,BATA,BATB
2404.....COMMON/MESH/RR(MT),ZZ(MT),DELRR(MT-1),DELZZ(MT-1),
2405.....1SR(IN2),SZ(JM1P1),DELR(IN2-1),DELZ(JM1P1-1)

```

```

2406.....COMMON/CONST/ N1,N2,N3,M1,M2,N1P1,N1P2,N2P1,N2P2,N2M1,
2407.....1N1PN2,N1PN2P,N1PN2M,M1P1,M1M1,M1P2,M2P1,M2M1,M2P2,MN,MNM1,MM,MMM1
2408.....COMMON/CUR/ NCUR,LOCUR(35),VALCUR(35)
2409.....COMMON/DIM/RSMA,RB,BB1,RLNGTH,ZLNGTH
2410.....COMMON/CDEFAD/CDEFR1(IN2-1,JM1P1-1),CDEFR2(IN2-1,JM1P1-1),
2411.....2CDEFR3(IN2-1,JM1P1-1),CDEFZ1(IN2-1,JM1P1-1),
2412.....3CDEFZ2(IN2-1,JM1P1-1),CDEFZ3(IN2-1,JM1P1-1)
2413.....COMMON/CONTOUR/ FLUXRZ(NT,MT),RAXIS(NT),ZAXIS(MT)
2414.....COMMON/VAR2/FR,FK,CV,C,Q,HU,VN,BT,TMAX,PTIME1,PTIME2,TIME,
2415.....1.TBIS1,TBIS2,
2416.....1BMIN,SGO,CDE,MW,MP,MC,NCURV,NCHECK,BTOR,DN,TEMP,
2417.....2CUR1MAX,CUR2MAX,CUR3MAX,CURTOR,RSPINCH,RBPINCH
2418.....COMMON/WSEM/ S(N,(-4,4))
2419.....COMMON/WLCM2/ GA(IN2,JM1P1),GB(IN2,JM1P1),GC(IN2,JM1P1),
2420.....1GD(IN2,JM1P1),GE(IN2,JM1P1),GF(IN2,JM1P1),VSEM1(4,IN2,JM1P1),
2421.....2VSEM2(4,IN2,JM1P1),APSI(NT,MT)
2422.....COMMON/LCM1CGG/ MTRX(N,(-4,4)),LU(N,(-4,4)),X(N),Y(N)
2423.....REAL MTRX,LU
2424.....DIMENSION AA(4*(IN2-1),12),BB(4,4)
2425.....OPTIMIZE
2426.....TMALL=1.0E-30
2427.....K2=3
2428.....K=1
2429.....SW=1.0
2430.....IF (TIME.GT.TBIS1) SW=1.0-2.0*BT/(BY*BY*TBIS2)
2431.....CALL QACOPY(VSEM1,W(1,1,1,1),4*NS)
2432.....CALL QACOPY(VSEM2,W(1,1,1,2),4*NS)
2433.....CALL QACOPY(GA,EI(1,1,1),NS)
2434.....CALL QACOPY(GB,EI(1,1,2),NS)
2435.....DO 38 J=JB,JE
2436.....DO 39 I=IB,IE
2437.....HON=0.4
2438.....IF (VSEM1(1,I,J).LT.0.3) HON=2.0
2439 C..... ADD MORE DISSIPATION TO THE UPPER RIGHT HAND CORNER.
2440 COM..... IF (TIME.GT.TBIS1.AND.I.GE.5.AND.J.GE.8) HON=2.0+HON
2441.....888 F(1,I,J,K2)=VSEM1(1,I,J)
2442.....1-BATA/SR(I)*(SR(I+1)+VSEM1(2,I+1,J)-
2443.....2SR(I-1)+VSEM1(2,I-1,J))+F(1,I,J,K)+F(1,I,J,K+1)
2444.....6+HON/4.0+0.5*BT/BX*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)/
2445.....7/VSEM1(1,I,J))*(VSEM1(1,I+1,J)-VSEM1(1,I,J))-ABS(VSEM1(2,I,J)/VSEM

```



```

2446.....91(1,I,J)
2447.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))+(VSEM1(1,I,J)-VSEM1(1,I-1,J))
2448.....6+HDM/4.+.5*BT/BY*(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
2449.....7/VSEM1(1,I,J))+(VSEM1(1,I,J+1)-VSEM1(1,I,J))-ABS(VSEM1(3,I,J)/VSEM
2450.....91(1,I,J)
2451.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))+(VSEM1(1,I,J)-VSEM1(1,I,J-1))
2452.....F(2,I,J,K2)=SW+VSEM1(2,I,J)+.5*(1.0-SW)*(VSEM1(2,I,J-1)+VSEM1(2,I
2453.....9,J+1))
2454.....1-BATA/SR(I)+(SR(I+1)+(VSEM1(2,I+1,J)+.2
2455.....2/VSEM1(1,I+1,J)+(FK-1.)*VSEM1(1,I+1,J)+GA(I+1,J))-SR(I-1)+
2456.....3(VSEM1(2,I-1,J)+.2/VSEM1(1,I-1,J)+(FK-1.)*VSEM1(1,I-1,J)
2457.....4+GA(I-1,J))+F(2,I,J,K)+F(2,I,J,K+1)
2458.....6+HDM/4.+.5*BT/BX*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)
2459.....7/VSEM1(1,I,J))+(VSEM1(2,I+1,J)-VSEM1(2,I,J))-ABS(VSEM1(2,I,J)/VSEM
2460.....91(1,I,J)
2461.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))+(VSEM1(2,I,J)-VSEM1(2,I-1,J))
2462.....6+HDM/4.+.5*BT/BY*(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
2463.....7/VSEM1(1,I,J))+(VSEM1(2,I,J+1)-VSEM1(2,I,J))-ABS(VSEM1(3,I,J)/VSEM
2464.....91(1,I,J)
2465.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))+(VSEM1(2,I,J)-VSEM1(2,I,J-1))
2466.....F(3,I,J,K2)=SW+VSEM1(3,I,J)+.5*(1.0-SW)*(VSEM1(3,I,J-1)+VSEM1(3,I
2467.....9,J+1))
2468.....1-BATA/SR(I)+(SR(I+1)+VSEM1(2,I+1,J)+
2469.....2VSEM1(3,I+1,J)/VSEM1(1,I+1,J)-SR(I-1)+VSEM1(2,I-1,J)+VSEM1(3,I-1,J
2470.....9)
2471.....3/VSEM1(1,I-1,J))+F(3,I,J,K)+F(3,I,J,K+1)
2472.....6+HDM/4.+.5*BT/BX*(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)
2473.....7/VSEM1(1,I,J))+(VSEM1(3,I+1,J)-VSEM1(3,I,J))-ABS(VSEM1(2,I,J)/VSEM
2474.....91(1,I,J)
2475.....8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))+(VSEM1(3,I,J)-VSEM1(3,I-1,J))
2476.....6+HDM/4.+.5*BT/BY*(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
2477.....7/VSEM1(1,I,J))+(VSEM1(3,I,J+1)-VSEM1(3,I,J))-ABS(VSEM1(3,I,J)/VSEM
2478.....91(1,I,J)
2479.....8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))+(VSEM1(3,I,J)-VSEM1(3,I,J-1))
2480.....F(4,I,J,K2)=SW+VSEM1(4,I,J)+.5*(1.0-SW)*(VSEM1(4,I,J-1)+VSEM1(4,I
2481.....9,J+1))
2482.....1-BATA/SR(I)+(SR(I+1)+VSEM1(2,I+1,J)
2483.....2+VSEM1(4,I+1,J)/VSEM1(1,I+1,J)-SR(I-1)+VSEM1(2,I-1,J)+VSEM1(4,I-1,
2484.....9J)
2485.....1/VSEM1(1,I-1,J))+F(4,I,J,K)+F(4,I,J,K+1)

```

```

2486 ..... 6+HON/4.♦0.5♦BT/BX♦(ABS(VSEM1(2,I+1,J)/VSEM1(1,I+1,J)-VSEM1(2,I,J)
2487 ..... 7/VSEM1(1,I,J))♦(VSEM1(4,I+1,J)-VSEM1(4,I,J))-ABS(VSEM1(2,I,J)/VSEM
2488 ..... 91(1,I,J)
2489 ..... 8-VSEM1(2,I-1,J)/VSEM1(1,I-1,J))♦(VSEM1(4,I,J)-VSEM1(4,I-1,J))
2490 ..... 6+HON/4.♦0.5♦BT/BY♦(ABS(VSEM1(3,I,J+1)/VSEM1(1,I,J+1)-VSEM1(3,I,J)
2491 ..... 7/VSEM1(1,I,J))♦(VSEM1(4,I,J+1)-VSEM1(4,I,J))-ABS(VSEM1(3,I,J)/VSEM
2492 ..... 91(1,I,J)
2493 ..... 8-VSEM1(3,I,J-1)/VSEM1(1,I,J-1))♦(VSEM1(4,I,J)-VSEM1(4,I,J-1))
2494 ..... 39. CONTINUE
2495 ..... F(2,IB,J,K2)=F(2,IB,J,K2)-2.0♦BATA♦SR(IB-1)/SR(IB)♦
2496 ..... 1VSEM1(2,IB-1,J)♦♦2/VSEM1(1,IB-1,J)
2497 ..... F(2,IE,J,K2)=F(2,IE,J,K2)+2.0♦BATA♦SR(IE+1)/SR(IE)♦
2498 ..... 1VSEM1(2,IE+1,J)♦♦2/VSEM1(1,IE+1,J)
2499 ..... 38. CONTINUE
2500 ..... DO 113 I=IB,IE
2501 ..... F(3,I,JE,K2)=F(3,I,JE,K2)-(FK-1.)♦VSEM2(1,I,JB)♦GB(I,JB)♦BATB
2502 ..... F(1,I,JE,K2)=F(1,I,JE,K2)-BATB♦VSEM2(3,I,JE+1)
2503 ..... F(2,I,JE,K2)=F(2,I,JE,K2)-BATB♦VSEM2(2,I,JE+1)♦
2504 ..... 1VSEM2(3,I,JE+1)/VSEM2(1,I,JE+1)
2505 ..... F(3,I,JE,K2)=F(3,I,JE,K2)+BATB♦VSEM2(3,I,JE+1)♦♦2/VSEM2(1,I,JE+1)
2506 ..... F(4,I,JE,K2)=F(4,I,JE,K2)-BATB♦VSEM2(3,I,JE+1)♦
2507 ..... 1VSEM2(4,I,JE+1)/VSEM2(1,I,JE+1)
2508 ..... DO 114 J=JS,JP
2509 ..... IF (J.EQ. JS) GO TO 115
2510 ..... AA(4♦J-3,1)=0.
2511 ..... AA(4♦J-3,2)=0.
2512 ..... AA(4♦J-3,3)=-BATB
2513 ..... AA(4♦J-3,4)=0.
2514 ..... AA(4♦J-2,1)=0.
2515 ..... AA(4♦J-2,2)=-BATB♦(VSEM2(3,I,J)/VSEM2(1,I,J))
2516 ..... AA(4♦J-2,3)=0.
2517 ..... AA(4♦J-2,4)=0.
2518 ..... AA(4♦J-1,1)=-BATB♦((FK-1.)♦GB(I,J))
2519 ..... AA(4♦J-1,2)=0.
2520 ..... AA(4♦J-1,3)=-BATB♦(VSEM2(3,I,J)/VSEM2(1,I,J))
2521 ..... AA(4♦J-1,4)=0.
2522 ..... AA(4♦J,1)=0.
2523 ..... AA(4♦J,2)=0.
2524 ..... AA(4♦J,3)=0.
2525 ..... AA(4♦J,4)=-BATB♦(VSEM2(3,I,J)/VSEM2(1,I,J))

```

```

2526...115 CONTINUE
2527...IF (J.EQ.JP) GO TO 116
2528...AA(4+J-3,9)=0:
2529...AA(4+J-3,10)=0:
2530...AA(4+J-3,11)=+BATB
2531...AA(4+J-3,12)=0:
2532...AA(4+J-2,9)=0:
2533...AA(4+J-2,10)=+BATB*(WSEM2(3,I,J+2)/WSEM2(1,I,J+2))
2534...AA(4+J-2,11)=0:
2535...AA(4+J-2,12)=0:
2536...AA(4+J-1,9)=+BATB*((FK-1.)+GB(I,J+2))
2537...AA(4+J-1,10)=0:
2538...AA(4+J-1,11)=+BATB*(WSEM2(3,I,J+2)/WSEM2(1,I,J+2))
2539...AA(4+J-1,12)=0:
2540...AA(4+J,9)=0:
2541...AA(4+J,10)=0:
2542...AA(4+J,11)=0:
2543...AA(4+J,12)=+BATB*(WSEM2(3,I,J+2)/WSEM2(1,I,J+2))
2544...116 CONTINUE
2545...AA(4+J-3,5)=1:
2546...AA(4+J-3,6)=0:
2547...AA(4+J-3,7)=0:
2548...AA(4+J-3,8)=0:
2549...AA(4+J-2,5)=-BT/(2.*SR(I))+*(FK-1.0)+GB(I,J+1)
2550...AA(4+J-2,6)=1:
2551...AA(4+J-2,7)=0:
2552...AA(4+J-2,8)=-BT/(2.*SR(I))+WSEM2(4,I,J+1)/WSEM2(1,I,J+1)
2553...AA(4+J-1,5)=0:
2554...AA(4+J-1,6)=0:
2555...AA(4+J-1,7)=1:
2556...AA(4+J-1,8)=0:
2557...AA(4+J,5)=0:0
2558...AA(4+J,6)=0:0
2559...AA(4+J,7)=0:
2560...AA(4+J,8)=1:0+BT/(2.*SR(I))+WSEM2(2,I,J+1)/WSEM2(1,I,J+1)
2561...114 CONTINUE
2562...CHECK REFLECTION BOUNDARY CONDITIONS ON Z=0 CAREFULLY
2563...DO 207 L=1,2
2564...DO 208 LL=9,12
2565...AA(L,LL)=2.*AA(L,LL)

```

```

2566.....AA(L+2,LL)=0.
2567...208 CONTINUE
2568...207 CONTINUE
2569 C.....MODIFICATIONS.....
2570 C
2571.....DO 499 LI=1,4
2572.....DO 499 LJ=5,12
2573.....IF (ABS(AA(LI,LJ)).LT.TMALL) AA(LI,LJ)=0.
2574...499 CONTINUE
2575 C
2576.....DO 498 LJ=6,12
2577...498 AA(1,LJ)=AA(1,LJ)/AA(1,5)
2578 C
2579.....DO 500 LI=2,4
2580...500 AA(LI,6)=AA(LI,6)-AA(LI,5)*AA(1,6)
2581 C
2582.....DO 501 LI=5,8
2583...501 AA(LI,2)=AA(LI,2)-AA(LI,1)*AA(1,6)
2584 C
2585.....DO 502 LJ=7,12
2586...502 AA(2,LJ)=(AA(2,LJ)-AA(2,5)*AA(1,LJ))/AA(2,6)
2587 C
2588.....DO 503 LI=3,4
2589.....DO 504 LK=1,2
2590...504 AA(LI,7)=AA(LI,7)-AA(LI,4+LK)*AA(LK,7)
2591...503 CONTINUE
2592 C
2593.....DO 505 LI=5,8
2594.....DO 506 LK=1,2
2595...506 AA(LI,3)=AA(LI,3)-AA(LI,LK)*AA(LK,7)
2596...505 CONTINUE
2597 C
2598.....DO 507 LJ=8,12
2599.....DO 508 LK=1,2
2600...508 AA(3,LJ)=AA(3,LJ)-AA(3,4+LK)*AA(LK,LJ)
2601...AA(3,LJ)=AA(3,LJ)/AA(3,7)
2602...507 CONTINUE
2603 C
2604.....DO 509 LK=1,3
2605...509 AA(4,8)=AA(4,8)-AA(4,4+LK)*AA(LK,8)

```

```

2606 C
2607 ..... DD 510 LI=5,8
2608 ..... DD 511 LK=1,3
2609 ..... 511 AA(LI,4)=AA(LI,4)-AA(LI,LK)*AA(LK,8)
2610 ..... 510 CONTINUE
2611 C
2612 ..... DD 512 LJ=9,12
2613 ..... DD 513 LK=1,3
2614 ..... 513 AA(4,LJ)=AA(4,LJ)-AA(4,4+LK)*AA(LK,LJ)
2615 ..... AA(4,LJ)=AA(4,LJ)/AA(4,8)
2616 ..... 512 CONTINUE
2617 C
2618 ..... F(1,I,JB,K+2)=F(1,I,JB,K+2)/AA(1,5)
2619 ..... DD 520 L=2,4
2620 ..... LE=L-1
2621 ..... DD 521 LK=1,LE
2622 ..... IF (ABS(AA(L,4+LK)).LT.TMALL) AA(L,4+LK)=0
2623 ..... IF (ABS(F(LK,I,JB,K+2)).LT.TMALL) F(LK,I,JB,K+2)=0
2624 ..... 521 F(L,I,JB,K+2)=F(L,I,JB,K+2)
2625 ..... 1-AA(L,4+LK)*F(LK,I,JB,K+2)
2626 ..... F(L,I,JB,K+2)=F(L,I,JB,K+2)/AA(L,4+L)
2627 ..... 520 CONTINUE
2628 C
2629 C.....END OF MODIFICATIONS.....
2630 ..... JPP=4*JP
2631 ..... DD 117 J=5,JPP,4
2632 ..... DD 188 L=1,4
2633 ..... DD 189 LL=1,4
2634 ..... BB(L,LL)=0
2635 ..... DD 121 LLL=1,4
2636 ..... IF (ABS(AA(J+L-1,LLL)).LT.TMALL) AA(J+L-1,LLL)=0
2637 ..... IF (ABS(AA(J+LLL-5,8+LL)).LT.TMALL) AA(J+LLL-5,8+LL)=0
2638 ..... 121 BB(L,LL)=BB(L,LL)+AA(J+L-1,LLL)*AA(J+LLL-5,8+LL)
2639 ..... 189 CONTINUE
2640 ..... 188 CONTINUE
2641 ..... DD 122 L=1,4
2642 ..... 122 AA(J-1+L,5)=AA(J-1+L,5)-BB(L,1)
2643 ..... DD 123 L=1,3
2644 ..... 123 AA(J,5+L)=(AA(J,5+L)-BB(1,L+1))/AA(J,5)
2645 ..... IF (J.EQ.(JPP-3)) GO TO 924

```

```

2646.....DD·124·L=1,4
2647.....124·AA(J,8+L)=AA(J,8+L)/AA(J,5)
2648.....924·CONTINUE
2649.....DD·125·L=1,3
2650.....125·AA(J+L,6)=AA(J+L,6)-BB(L+1,2)-AA(J+L,5)♦AA(J,6)
2651.....DD·126·L=1,2
2652.....126·AA(J+1,6+L)=(AA(J+1,6+L)-BB(2,2+L)-AA(J+1,5)♦AA(J,6+L))/AA(J+1,6)
2653.....IF·(J·EQ·(JPP-3))·GO·TO·127
2654.....DD·128·L=1,4
2655.....AA(J+L+3,2)=AA(J+L+3,2)-AA(J+L+3,1)♦AA(J,6)
2656.....AA(J+1,8+L)=(AA(J+1,8+L)-AA(J+1,5)♦AA(J,8+L))/AA(J+1,6)
2657.....128·CONTINUE
2658.....127·CONTINUE
2659.....DD·129·L=1,2
2660.....129·AA(J+L+1,7)=AA(J+L+1,7)-BB(2+L,3)-AA(J+L+1,5)♦AA(J,7)-
2661.....1AA(J+L+1,6)♦AA(J+1,7)
2662.....AA(J+2,8)=(AA(J+2,8)-BB(3,4)-AA(J+2,5)♦AA(J,8)-AA(J+2,6)♦AA(J+1,8)
2663.....2)/AA(J+2,7)
2664.....IF·(J·EQ·(JPP-3))·GO·TO·130
2665.....DD·131·L=1,4
2666.....DD·132·LL=1,2
2667.....AA(J+L+3,3)=AA(J+L+3,3)-AA(J+L+3,LL)♦AA(J+LL-1,7)
2668.....AA(J+2,8+L)=AA(J+2,8+L)-AA(J+2,4+LL)♦AA(J+LL-1,8+L)
2669.....132·CONTINUE
2670.....AA(J+2,8+L)=AA(J+2,8+L)/AA(J+2,7)
2671.....131·CONTINUE
2672.....130·CONTINUE
2673.....AA(J+3,8)=AA(J+3,8)-BB(4,4)-AA(J+3,5)♦AA(J,8)-AA(J+3,6)♦AA(J+1,8)
2674.....1-AA(J+3,7)♦AA(J+2,8)
2675.....IF·(J·EQ·(JPP-3))·GO·TO·133
2676.....DD·134·L=1,4
2677.....DD·135·LL=1,3
2678.....AA(J+L+3,4)=AA(J+L+3,4)-AA(J+L+3,LL)♦AA(J+LL-1,8)
2679.....AA(J+3,8+L)=AA(J+3,8+L)-AA(J+3,4+LL)♦AA(J+LL-1,8+L)
2680.....135·CONTINUE
2681.....AA(J+3,8+L)=AA(J+3,8+L)/AA(J+3,8)
2682.....134·CONTINUE
2683.....133·CONTINUE
2684.....DD·136·L=1,4
2685.....DD·137·LL=1,4

```

```

2686...137 F(L,I,(J+7)/4,K+2)=F(L,I,(J+7)/4,K+2)-AA(J+L-1,LL)*F(LL,I,(J+7)/4
2687...1-1,K+2)
2688...LE=L-1
2689...IF (L.EQ.1) GO TO 148
2690...DO 147 LL=1,LE
2691...F(L,I,(J+7)/4,K+2)=F(L,I,(J+7)/4,K+2)-AA(J+L-1,4+LL)*F(LL,I,(J+7)
2692...1/4,K+2)
2693...147 CONTINUE
2694...148 CONTINUE
2695...F(L,I,(J+7)/4,K+2)=F(L,I,(J+7)/4,K+2)/AA(J+L-1,5+L-1)
2696...136 CONTINUE
2697...117 CONTINUE
2698...W(4,I,JE,K+2)=F(4,I,JE,K+2)
2699...W(3,I,JE,K+2)=F(3,I,JE,K+2)-AA(JPP-1,8)*W(4,I,JE,K+2)
2700...W(2,I,JE,K+2)=F(2,I,JE,K+2)-AA(JPP-2,8)*W(4,I,JE,K+2)
2701...1-AA(JPP-2,7)*W(3,I,JE,K+2)
2702...W(1,I,JE,K+2)=F(1,I,JE,K+2)-AA(JPP-3,8)*W(4,I,JE,K+2)
2703...1-AA(JPP-3,7)*W(3,I,JE,K+2)-AA(JPP-3,6)*W(2,I,JE,K+2)
2704...DO 138 J=5,JPP,4
2705...JJ=JPP+5-J-4
2706...DO 139 L=1,4
2707...BB(1,L)=0
2708...DO 941 LL=1,4
2709...941 BB(1,L)=BB(1,L)+AA(JJ+1-L,8+LL)*W(LL,I,(JJ+4)/4+1,K+2)
2710...139 CONTINUE
2711...W(4,I,(JJ+4)/4,K+2)=F(4,I,(JJ+4)/4,K+2)-BB(1,1)
2712...W(3,I,(JJ+4)/4,K+2)=F(3,I,(JJ+4)/4,K+2)-BB(1,2)
2713...1-AA(JJ-1,8)*W(4,I,(JJ+4)/4,K+2)
2714...W(2,I,(JJ+4)/4,K+2)=F(2,I,(JJ+4)/4,K+2)-BB(1,3)
2715...1-AA(JJ-2,8)*W(4,I,(JJ+4)/4,K+2)-AA(JJ-2,7)*W(3,I,(JJ+4)/4,K+2)
2716...W(1,I,(JJ+4)/4,K+2)=F(1,I,(JJ+4)/4,K+2)-BB(1,4)
2717...1-AA(JJ-3,8)*W(4,I,(JJ+4)/4,K+2)-AA(JJ-3,7)*W(3,I,(JJ+4)/4,K+2)
2718...2-AA(JJ-3,6)*W(2,I,(JJ+4)/4,K+2)
2719...138 CONTINUE
2720...113 CONTINUE
2721...RETURN
2722...END

```