ORNL/TM--7871

DE82 008142

ORNL/TM-7871
Dist. Category UC-20 a,d,g

Contract No. W-7405-eng-26

FUSION ENERGY DIVISION

VMOMS: A COMPUTER CODE FOR FINDING MOMENT SOLUTIONS TO THE GRAD-SHAFRANOV EQUATION

L. L. Lao
TRW, Redondo Beach, CA 90278

R. M. Wieland
Computer Sciences

W. A. Houlberg, S. P. Hirshman
Fusion Energy Division

Date Published - February 1982

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

Abstract

A code VMOMS is described which finds approximate solutions to the Grad-Shafranov equation describing scalar pressure-balance equilibria for axisymmetric tokamak plasmas. A Fourier series expansion of the flux surface coordinates (R,Z) is made in terms of two new coordinates (ρ,θ), and the resulting equation is conveniently reduced to a system of ordinary differential equations (ODE's) using a variational principle. The solution of these simple equations with pressure and current as driving functions, yields, in principle, a complete description of the equilibrium. Complete axisymmetry is assumed, as well as up-down symmetry about the toroidal midplane.

1. Introduction

Repeated solutions of the two-dimensional (2-D) Grad-Shafranov equation for the evolving flux surface geometry in a tokamak can significantly increase the computer time and storage requirement of a transport code. A computationally efficient method to find approximate solutions to the Grad-Shafranov equation has been developed [1]. The flux surface coordinates (R,Z) are expanded in Fourier series in a poloidal angle θ that increases by 2π the short way around the torus (Fig. 1):

$$R(\rho,\theta) = R_0(\rho) - R_1(\rho)\cos\theta + \sum_{n=2}^{n_R} R_n(\rho)\cos n\theta \ , \qquad (1.1)$$

and

$$Z(\rho,\theta) = \sum_{n=1}^{n_Z} Z_n(\rho)\sin n\theta = E(\rho) \sum_{n=1}^{n_R} R_n(\rho)\sin n\theta \ , \qquad (1.2)$$

where ρ is a flux surface label running from the magnetic axis 0 to the plasma minor radius a, and the flux surfaces are assumed to have up-down symmetry. Also, the poloidal angle θ has been chosen such that $Z_n(\rho) = E(\rho)R_n(\rho)$. The amplitude functions $R_0(\rho)$, $R_1(\rho)$, and $E(\rho)$ describe the shift, the minor radius, and the ellipticity of the flux surfaces, respectively, and the amplitude $R_2(\rho)$ describes their triangularity. $R_0(a) \equiv R_c$ is the major radius of the torus. In VMOMS, R_1 is chosen to be the flux surface label ρ. A few terms ($n \leq 3$) in the Fourier series are generally sufficient to describe many plasma equilibria, including those for high-beta, strongly D-shaped plasmas. Thus, the problem of finding 2-D magnetohydrodynamic (MHD) equilibria

s reduced to that of solving a few ordinary differential equations (ODE's) for the amplitude functions $R_n(\rho)$ and $E(\rho)$.

2. Moment equations

The ODE's describing $R_n(\rho)$ and $E(\rho)$ can be conveniently obtained using a variational principle [1]. They can be shown to be moments of the Grad-Shafranov equation with basis functions M_{R_n} and M_E:

$$\langle M_{R_n} \tilde{G} \rangle = 0 , \tag{2.1}$$

and

$$\langle M_E \tilde{G} \rangle = 0 , \tag{2.2}$$

where $\langle\ \rangle$ is the poloidal-angle-averaging operator defined for any scalar A as

$$\langle A \rangle = \int_0^{2\pi} \frac{d\theta}{2\pi} A , \tag{2.3}$$

$$M_{R_0} = RZ_\theta , \tag{2.4}$$

$$M_{R_1} = R(ER_\theta \sin\theta + Z_\theta \cos\theta) , \tag{2.5}$$

$$M_{R_n} = R(ER_\theta \sin n\theta - Z_\theta \cos n\theta) \quad (n \quad 2), \tag{2.6}$$

and

$$M_E = \sum_{n=1}^{n_R} R_n R R_\theta \sin n\theta . \tag{2.7}$$

The subscripts (ρ and θ) denote differentiation with respect to these variables, and $\tilde{G}$ is the Grad-Shafranov operator in (ρ,θ) coordinates:

$$\tilde{G} = \frac{I^2(\rho)}{\pi c^2 \sqrt{g}\,\langle g_{\theta\theta}/\sqrt{g}\rangle}\left(\frac{\partial}{\partial\rho}\frac{g_{\theta\theta}}{\sqrt{g}\langle g_{\theta\theta}/\sqrt{g}\rangle} - \frac{\partial}{\partial\theta}\frac{g_{\rho\theta}}{\sqrt{g}\langle g_{\theta\theta}/\sqrt{g}\rangle}\right)$$

$$+ \frac{II'(\rho)}{\pi c^2 \langle g_{\theta\theta}/\sqrt{g}\rangle}\left(\frac{g_{\theta\theta}}{g\langle g_{\theta\theta}/\sqrt{g}\rangle} - \frac{1}{g_{\phi\phi}\langle\sqrt{g}/g_{\phi\phi}\rangle}\right)$$

$$+ p'(\rho)\left(1 - \frac{\langle\sqrt{g}\rangle}{g_{\phi\phi}\langle\sqrt{g}/g_{\phi\phi}\rangle}\right) . \qquad (2.8)$$

Here, $p(\rho)$ is the plasma pressure, $I(\rho)$ is the toroidal plasma current enclosed by a flux surface labeled ρ, $\sqrt{g}$ is the (3-D) Jacobian of the transformation from (R, ϕ, Z) to (ρ, θ, ϕ) coordinates (ϕ is the ignorable toroidal angle),

$$\sqrt{g} = R\tau , \qquad (2.9)$$

and

$$\tau = R_\theta Z_\rho - R_\rho Z_\theta . \qquad (2.10)$$

For $(\mu,\nu) = (\rho,\theta)$, $g_{\mu\nu}$ are the elements of the metric tensor

$$g_{\mu\nu} = R_\mu R_\nu + Z_\mu Z_\nu , \qquad (2.11)$$

and

$$g_{\phi\phi} = R^2 . \qquad (2.12)$$

The moment equations (2.1) and (2.2) can be written explicitly as

$$d^y_{10} r_0''(x) + \sum_{n=2}^{n_R} [d^y_{1n} - E(x) d^y_{2n}] r_n''(x)$$

$$- \sum_{n=1}^{n_R} r_n d^y_{2n} E''(x) = -\beta_{p_0} d^y_3 \frac{\hat{p}'(x)}{\hat{I}^2(x)} - d^y_4$$

$$+ d^y_5 + \sum_{n=1}^{n_R} 2 d^y_{2n} E'(x) r_n'(x) \quad , \tag{2.13}$$

where

$$x = \frac{\rho}{a} \quad , \tag{2.14}$$

a is the flux surface label of the outermost flux surface,

$$r_n(x) = \frac{R_n(\rho)}{a} \quad , \tag{2.15}$$

$$\hat{p}(x) = \frac{p(\rho)}{p_0} \quad , \tag{2.16}$$

$$\hat{I}(x) = \frac{I(\rho)}{I_0} \quad , \tag{2.17}$$

$$\beta_{p_0} = \frac{\pi c^2 p_0 a^2}{I_0^2} \quad , \tag{2.18}$$

$$d^y_{1n} = \langle (GS_{1n} - GS_8 \frac{\langle \hat{\tau} GS_{1n} \rangle}{\langle \hat{\tau} GS_8 \rangle}) M_y \rangle \quad , \tag{2.19}$$

$$d^y_{2n} = \langle (GS_{2n} - GS_8 \frac{\langle \hat{\tau} GS_{2n} \rangle}{\langle \hat{\tau} GS_8 \rangle}) M_y \rangle \quad , \tag{2.20}$$

$$d_3^y = \langle \hat{\tau} GS_8 \rangle^2 \left\langle \left(GS_3 - GS_4 \frac{\langle \hat{\tau} GS_3 \rangle}{\langle \hat{\tau} GS_4 \rangle} \right) M_y \right\rangle , \tag{2.21}$$

$$d_4^y = \left\langle \left(SS_2 - GS_4 \left\langle \frac{\hat{\tau} SS_2}{\hat{\tau} GS_4} \right\rangle \right) M_y \right\rangle , \tag{2.22}$$

and

$$d_5^y = CS_1 \left\langle \left(GS_8 - GS_4 \frac{\langle \hat{\tau} GS_8 \rangle}{\langle \hat{\tau} GS_4 \rangle} \right) M_y \right\rangle . \tag{2.23}$$

Here, M_y are the appropriate basis functions M_{R_n} and M_E, and

$$GS_{1n} = \frac{z_\theta \hat{g}_{\theta\theta}}{r\hat{\tau}^3} \cos n\theta , \tag{2.24}$$

$$GS_{2n} = \frac{r_\theta \hat{g}_{\theta\theta}}{r\hat{\tau}^3} \sin n\theta , \tag{2.25}$$

$$GS_3 = r , \tag{2.26}$$

$$GS_4 = 1/r , \tag{2.27}$$

$$GS_5 = \frac{z_\theta}{r^2\hat{\tau}} , \tag{2.28}$$

$$GS_6 = \frac{1}{r\hat{\tau}^3} \left[\hat{g}_{\theta\theta} (r_x z_{x\theta} - r_{x\theta} z_x) + \hat{g}_{\rho\theta} (r_{\theta\theta} z_x + r_\theta z_{x\theta} - r_{x\theta} z_\theta - r_x z_{\theta\theta}) \right] , \tag{2.29}$$

$$GS_7 = \frac{1}{r\hat{\tau}^2} \left(r_\theta r_{x\theta} + z_\theta z_{x\theta} - r_x r_{\theta\theta} - z_x z_{\theta\theta} \right) . \tag{2.30}$$

$$GS_8 = \frac{\hat{g}_{\theta\theta}}{\hat{r}_\tau^2} , \tag{2.31}$$

$$CS_1 = \frac{1}{\langle \hat{\tau} GS_8 \rangle} \left\langle \frac{\hat{g}_{\theta\theta x}}{\sqrt{\hat{g}}} - \frac{\hat{g}_{\theta\theta} r_x}{r^2 \hat{\tau}} + \frac{(r_x z_{x\theta} - r_{x\theta} z_x)}{\hat{r}_\tau^2} \hat{g}_{\theta\theta} \right\rangle , \tag{2.32}$$

$$SS_2 = GS_5 + GS_6 + GS_7 + GS_8 \frac{\hat{I}'(x)}{\hat{I}(x)} , \tag{2.33}$$

and

$$\hat{g}_{\theta\theta x} = 2(r_\theta r_{x\theta} + z_\theta z_{x\theta}) , \tag{2.34}$$

where $\hat{\tau}$, $\sqrt{\hat{g}}$, $\hat{g}_{\mu\nu}$, r, and z are the appropriate normalized quantities found in eqs. (1.1)-(1.2) and (2.9)-(2.11).

Given the plasma pressure and the toroidal current, $p(\rho)$ and $I(\rho)$, respectively, eq. (2.13) defines a set of coupled second-order ODE's for the Fourier amplitude functions $R_n(\rho)$ and $E(\rho)$. VMOMS is written to solve the set of moment equations (2.13) with the boundary conditions corresponding to a fixed plasma boundary using a shooting technique.

3. Derived quantities

The solution of eq. (2.13) for $r_n(x)$ and $E(x)$ leads directly to a full specification of the flux surfaces $R(\rho,\theta)$ and $Z(\rho,\theta)$, eqs. (1.1)-(1.2), and thus the metric coefficients $g_{\mu\nu}$, eqs. (2.11)-(2.12).

It is then straightforward to define a group of derived parameters that characterize the equilibrium.

3.1. Volume

The differential volume is given by

$$dV = \sqrt{g}\, d\rho\, d\theta\, d\phi \ . \tag{3.1}$$

Integrating out the angle variables leads to

$$\frac{dV}{d\rho} = 4\pi^2 \langle \sqrt{g} \rangle \ . \tag{3.2}$$

3.2. Flux tube surface area

The surface area enclosing a flux tube generated by rotating a given flux contour completely around 2π radians in the ϕ direction is given by

$$S = 4\pi^2 \langle \sqrt{g} |\nabla\rho| \rangle \ . \tag{3.3}$$

3.3. Poloidal flux

The surface average of the toroidal component of Ampere's Law gives

$$\chi'(\rho) = -\frac{(4\pi/c) I(\rho)}{\langle g_{\theta\theta}/\sqrt{g} \rangle} \ , \tag{3.4}$$

where χ is defined by

$$|\nabla\chi(\rho)| = 2\pi R B_p \ . \tag{3.5}$$

3.4. Poloidal current

Similarly, the poloidal current enclosed between the axis of symmetry and the magnetic surface ρ,

$$F(\rho) = \frac{c}{2} R B_T(R) \ , \tag{3.6}$$

can be written in terms of the moments by surface averaging the pressure balance equation

$$\langle\sqrt{g}\tilde{G}\rangle = 0 \tag{3.7}$$

to give

$$F^2(\rho) = F_0^2 \left\{1 + \frac{1}{q_0^2 A^4} \int_\rho^a \frac{d\rho}{\langle\sqrt{g}/g_{\phi\phi}\rangle} \left[\frac{2\beta p_0 \langle\sqrt{g}\rangle}{p_0 a^2} p'(\rho) + \frac{1}{I_0^2 \langle g_{\theta\theta}/\sqrt{g}\rangle} I^{2\prime}(\rho)\right]\right\} \ , \tag{3.8}$$

where

$$F_0 = \frac{c}{2} R_c B_T(R_c) \ , \tag{3.9}$$

$$q_0 = \frac{F_0}{I_0 A^2} \, , \tag{3.10}$$

and

$$A = \frac{R_c}{a} \, . \tag{3.11}$$

3.5. Toroidal flux

$$\Phi(\rho) = \frac{4\pi}{c} \int_0^\rho F(\rho) \left\langle \frac{\sqrt{g}}{g_{\theta\theta}} \right\rangle d\rho \, . \tag{3.12}$$

3.6. Toroidal current density

$$j_\phi(\rho,\theta) = -j_{\phi_0} \frac{\langle g_{\theta\theta}/\sqrt{g} \rangle I_0}{I(\rho)} \left\{ \left[\frac{\beta_{T_0} a p'(\rho) \left[\frac{R(\rho,\theta)}{R_c} \right]^2}{p_0} \right.\right.$$

$$\left.\left. + \frac{aFF'(\rho)}{F_0^2} \right] \right\} \frac{R_c}{R(\rho,\theta)} \, , \tag{3.13}$$

where

$$j_{\phi_0} = \frac{F_0^2}{2\pi a R_c I_0} \tag{3.14}$$

and

$$\beta_{T_0} = \frac{\pi p_0 R_c^2 c^2}{F_0^2} \, . \tag{3.15}$$

3.7. Equivalent current density

$$\tilde{j}(\rho) = \frac{\langle \sqrt{g}\ j_\phi(\rho,\theta)/R \rangle}{\langle \sqrt{g}/R \rangle} \, , \tag{3.16}$$

where

$$I(\rho) = \int_0^\rho \tilde{j}(\rho) dA \tag{3.17}$$

with $dA = 2\pi \langle \tau \rangle d\rho$. (3.18)

3.8. Safety factor

The safety factor q is defined in the usual way:

$$q(\rho) = \frac{d\phi}{d\chi} = q_0 A^2 \frac{\hat{F}}{I} \left\langle \frac{\sqrt{g}}{g_{\phi\phi}} \right\rangle \left\langle \frac{g_{\theta\theta}}{\sqrt{g}} \right\rangle \, . \tag{3.19}$$

4. Numerical formulation

The set of moment equations (2.13) is a coupled system of second-order ODE's for the unknowns $r_0(x)$, $r_n(x)$ $(n > 2)$, and $E(x)$. A solution requires two boundary conditions for each moment equation. The so-called "fixed boundary condition" provides one of these per equation by specifying the shape of the outermost flux surface:

$$r_n(a) = r_{na} \quad (n = 0,2, \ldots) , \tag{4.1}$$

$$E(a) = E_a . \tag{4.2}$$

The remaining boundary conditions are obtained [1] by requiring that the poloidal magnetic flux $\chi(R,Z)$ be analytic at the magnetic axis, i.e., that the flux contours be concentric ellipses near the magnetic axis. They are given by

$$r_0(0) = r_m , \tag{4.3}$$

$$E(0) = E_m , \tag{4.4}$$

$$r_n(0) = 0 \quad (n = 2, \ldots) , \tag{4.5}$$

and

$$r_0'(0) = 0 , \tag{4.6}$$

$$E'(0) = 0 , \tag{4.7}$$

$$r_n'(0) = 0 \quad (n = 2, \ldots) . \tag{4.8}$$

The conditions (4.3)-(4.8) define $\rho = 0$ as a critical point of the system, and special numerical steps must be taken to avoid this singularity. The solution space in ρ is contracted from [0,a] to $[\delta_0 a, a]$, where δ_0 is a small positive number. The analytic properties [1] of the solutions of r_n, E near the origin $\rho = 0$ are then used to rewrite eqs. (4.3)-(4.5) as

$$r_0(\delta_0) = r_m - \lambda_1 \delta_0^2 , \tag{4.9}$$

$$r_n(\delta_0) = r_{n_0}(\delta_0^n)^2 \quad (n = 2, \ldots) , \tag{4.10}$$

and

$$E(\delta_0) = E_m + \lambda_2 \delta_0^2 , \tag{4.11}$$

where λ_1 and λ_2 are arbitrarily small positive constants. The solutions obtained are insensitive to the exact values of λ_1 and λ_2.

The corresponding boundary conditions on the derivatives are then

$$r_0'(\delta_0) = -2\lambda_1 \delta_0 , \tag{4.12}$$

$$r_n'(\delta_0) = 2n r_{n_0} \delta_0^{2n-1} \quad (n = 2, \ldots) , \tag{4.13}$$

$$E'(\delta_0) = 2\lambda_2 \delta_0 . \tag{4.14}$$

The set of the amplitude functions r_n and E_n is infinite. To model an equilibrium plasma at finite beta, it is useful to consider only a finite number of amplitude functions r_n. Our experience and that of others [2] indicate that the set r_0, r_2, E with $n_R = 2$ is sufficient for most cases. The resulting finite set of equations can be written as

$$\underline{\underline{A}}\ \underline{y}'' = \underline{b}\ , \tag{4.15}$$

where the elements of the matrix $\underline{\underline{A}}$,

$$\underline{\underline{A}} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \tag{4.16}$$

are functions of the unknown vector y,

$$\underline{y} = [r_0(x),\ E(x),\ r_2(x)]\ , \tag{4.17}$$

and its first derivatives. The vector $\underline{b}$ is a function of $\underline{y}$, $\underline{y}'$ and the driving functions p'(x), I(x), and I'(x).

To solve this system of equations (4.15), the well-known "shooting method" [3] is employed using Gladwell's code D02AGF [4], which is obtainable from the NAG library of mathematical software. The system is rewritten as a set of six first-order ODE's with the solution vectors defined now as explicit functions of a parameter vector $\tilde{p}$:

$$\tilde{y} = (y_1, y_1', y_2, y_2', y_3, y_3') = \tilde{y}(x;\tilde{p})\ , \tag{4.18}$$

$$\underline{\underline{\tilde{A}}}\,\tilde{y}' = \tilde{b}\ , \tag{4.19}$$

with the components of $\tilde{p}$ defined in terms of the six remaining unknown boundary conditions (cf. Table I). A solution of the ODE system is

obtained by solving a parameter estimation problem for $\tilde{\underline{p}}$ [5], where the solution $\tilde{\underline{y}}^*$ to $\tilde{\underline{\underline{A}}}\tilde{\underline{y}}^{*\prime} = \tilde{\underline{b}}$ is equivalent to finding the $\tilde{\underline{p}}^*$ that gives

$$\tilde{\underline{d}}(\tilde{\underline{p}}^*) = 0 \ , \tag{4.20}$$

where

$$\tilde{\underline{d}}(\tilde{\underline{p}}) = \tilde{\underline{y}}_{\rightarrow}(1;\tilde{\underline{p}}) - \tilde{\underline{y}}_b(1;\tilde{\underline{b}}) \ . \tag{4.21}$$

$\tilde{\underline{y}}_{\rightarrow}(1;\tilde{\underline{p}})$ is the solution at x = 1 obtained by integrating to the right from $x = \delta_o$; $\tilde{\underline{y}}_b(1;\tilde{\underline{p}})$ is the boundary condition vector at x = 1.

The solution of the nonlinear system of equations (4.20) is obtained by a modified Newton iteration that formally calculates the kth iterate of $\tilde{\underline{p}}$ from

$$\tilde{\underline{p}}^{(k+1)} = \tilde{\underline{p}}^{(k)} - \{\underline{\underline{J}}[\tilde{\underline{p}}^{(k)}]\}^{-1}\tilde{\underline{d}}[\tilde{\underline{p}}^{(k)}] \ , \tag{4.22}$$

where, formally,

$$[J(\tilde{\underline{p}})]_{ij} = \frac{\partial \tilde{d}_i}{\partial p_j} \ . \tag{4.23}$$

The right-hand term in eq. (4.22) is approximated by a solution to the linear system

$$\underline{\underline{J}}[\tilde{\underline{p}}^{(k)}]\underline{\delta p}^{(k)} = -\underline{d}[\tilde{\underline{p}}^{(k)}] \tag{4.24}$$

so that the new iterate becomes

$$\tilde{\underline{p}}^{(k+1)} = \tilde{\underline{p}}^{(k)} + \lambda_k \delta\tilde{\underline{p}}^{(k)} \quad , \tag{4.25}$$

where $\lambda_k << 1$, chosen such that

$$\| \tilde{\underline{d}}[\tilde{\underline{p}}^{(k+1)}] \| < \| \tilde{\underline{d}}[\tilde{\underline{p}}^{(k)}] \| \quad . \tag{4.26}$$

$\underline{\underline{J}}[\tilde{\underline{p}}^{(k)}]$ is obtained by forward differencing $\tilde{\underline{p}}$ and integrating $\tilde{\underline{y}}$, as in an initial value problem.

5. Program description

VMOMS was written as a package of modular subroutines, designed to be easily interfaced to any external driver. The specific application intended was for linkage with the 1-D plasma transport simulation code WHIST [6], where it would be called upon to solve repeatedly the Grad-Shafranov moment equations (2.13) for a given state of the evolving plasma. To exhibit its application here, we have included a simple driver main program, MDRIV, with the VMOMS package. This driver sets up the necessary input profiles and scalar parameters and generates a concise output summary listing of the results.

5.1. Main driving program MDRIV

Description

MDRIV is the driver program for the entire package. It is supplied for the purpose of illustrating the use of the VMOMS routines and for generating sample output for an arbitrary input case.

Input parameters relevant to machine geometry and magnetics, plasma profile shapes, and numerical software operation are read in via the NAMELIST file /MOMIN/ (cf. Table II).

After calling the MHDEQ, the main subroutine which returns the moment functions on a 25-point uniform grid defined in MDRIV, the driver calls three more subroutines for further processing of the solution:

(1) GPASMA, which calculates additional parameters of interest;

(2) PRTMEQ, which generates a summary output of the results, and

(3) PLTMEQ, which writes a plot file for postprocessing generation of plots.

The driving functions for pressure and current are provided directly to the shooting code by subroutines PRSFUN and CURFUN, respectively.

5.2. Subroutine MHDEQ

Use

```
SUBROUTINE MHDEQ(KMHDEQ,IPARAM,NRHO,XRO,XBTO,XPO,XAIO,XEO,XE1,XDO,RHO,
     SHIFT,ELONG,TRIANG,IFAIL)
DIMENSION RHO(NRHO),SHIFT(NRHO),ELONG(NRHO),TRIANG(NRHO)
```

where

INPUT

KMHDEQ = 1 solve only the R_0 equation; treat E and R_2 as quadratic driving functions

= 2 solve the R_0 and E equations; treat R_2 as a quadratic driving function

= 3 solve the R_0, E, and R_2 equations simultaneously

IPARAM = 0 construct initial guess for parameter vector $\underset{\sim}{p}$ internally

= 1 use $\underset{\sim}{p}$ from prior solution (if available) as initial guess

NRHO = dimension of the external grid, RHO

XRO = major radius (cm) of the plasma

XBTO = toroidal magnetic field (G)

XPO = central plasma pressure (eV/cm^3)

XAIO = toroidal current (A)

XEO = plasma boundary elongation*

XE1 = magnetic axis elongation (KMHDEQ = 1 only)

XDO = plasma boundary triangularity*

RHO = external grid (cm)

---

*Cf. GEOTRN for definitions.

OUTPUT

SHIFT =	shift profile (shift in centimeters of the contour center with respect to center of the outermost contour)
ELONG =	geometric elongation profile*
TRIANG =	geometric triangularity profile*
IFAIL =	error flag (cf. Table III)

SHIFT, ELONG, and TRIANG are all defined on the external grid RHO.

Description

MHDEQ serves as the driver for the shooting code D02AGF. On the first pass, it initializes all the scalar and vector parameters that are internal to the shooting package. In particular, the parameter vector PARAM is initialized here with universal values that possess sufficient generality to start most problems successfully toward an obtainable solution (cf. Table I).

All input parameters are converted, where appropriate, to normalized units (designated by lower case letters) by dividing by the minor radius a. The boundary values for elongation and triangularity are converted from geometric-unit definitions to moment-unit definitions in GEOTRN.

The call to D02AGF returns the solution vector $\tilde{p}$ in the array PARAM and the moment vectors $\underline{r}_0$, $\underline{E}$, and $\underline{r}_2$ in the array C, with the latter defined over a coarse 10-point contracted grid XP(I), XP(I) ε [XLO,1.0]. The vectors are converted back to geometrical units (GEOTRN) and interpolated back onto the external driver grid, RHO.

---

*Cf. GEOTRN for definitions.

5.3. Subroutine GEOTRN

Use

CALL GEOTRN(MODE,XIN,EIN,DIN,EOUT,DOUT)

where

INPUT

MODE = 1 to convert from geometrical $\tilde{E},\tilde{D}$ to moments E,D

= 2 to convert from moments E,D to geometrical $\tilde{E},\tilde{D}$

XIN = reduced minor radius (ρ/a) of the contour

EIN = input elongation

DIN = input triangularity

OUTPUT

EOUT = output elongation

DOUT = output triangularity

Description

Referring to Fig. 2, it is convenient to define a geometric elongation $\tilde{E}$ as

$$\tilde{E} = b/\rho \ , \tag{5.1}$$

where b is the maximum altitude of the contour above the midplane and ρ is the midplane minor radius of the contour as measured from the geometric center R_G along the midplane.

Similarly, define a geometrical triangularity $\tilde{D}$ as

$$\tilde{D} = -\delta/\rho \ , \tag{5.2}$$

where δ is the horizontal offset of the maximum altitude point on the contour from R_G.

Defining $\tilde{E}$ and $\tilde{D}$ in terms of the moment definitions of R and Z (1.1)-(1.2), then

$$\tilde{E} = \frac{E(\rho)[\rho \sin \theta_c + D(\rho)\sin 2\theta_c]}{\rho} , \tag{5.3}$$

and

$$\tilde{D} = \frac{D(\rho) + \rho \cos \theta_c - D(\rho)\cos 2\theta_c}{\rho} , \tag{5.4}$$

where θ_c is the critical angle at which Z is a maximum, i.e.,

$$\rho \cos \theta_c + 2D(\rho)\cos 2\theta_c = 0 . \tag{5.5}$$

In MODE = 1 operation, eqs. (5.4)-(5.5) are solved recursively for $D(\rho)$ and θ_c; $E(\rho)$ is then obtained from eq. (5.3).

In MODE = 2 operation, $\tilde{E}$ and $\tilde{D}$ are obtained straightforwardly from eqs. (5.3)-(5.5).

5.4. Subroutine PRSFUN

Use

```
SUBROUTINE PRSFUN(X,P,PP)
```

where

INPUT

X = the point $x \in [\delta_0, 1]$ at which $\hat{p}$ and $\hat{p}'$ are required

OUTPUT

P = $p(x)/p_0$

PP = $p'(x)/p_0$

Description

The pressure profile, defined on the external grid RHO, is provided by array PRES in COMMON/SWORK1/. For each new problem, cubic spline coefficients are generated (SPLAAN) on the initial pass through the subroutine, and the interpolated values of pressure (SEVAL) and its derivative (SPEVAL) are returned, both normalized by the peak pressure p_0.

5.5. Subroutine CURFUN

Use

SUBROUTINE CURFUN(X,AI,AIP)

where

INPUT

X = the point x at which $\hat{I}$ and $\hat{I}'$ are required.

OUTPUT

AI = $I(x)/I_0$

AIP = $I'(x)/I_0$

Description

The current profile, defined on the external grid RHO, is provided by array CURT in COMMON/SWORK1/. For each new problem, cubic spline coefficients are generated by SPLAAN on the initial pass through the

subroutine, and the interpolated values of current (SEVAL) and its derivative (SPEVAL) are returned, both normalized by the total current I_0.

5.6. Subroutine BCAUX

Use

```
SUBROUTINE BCAUX(GL,GR,PARAM)
DIMENSION GL(6),GR(6),PARAM(6)
```

where

INPUT

PARAM = parameter vector $\tilde{p}$ provided by D02AGF

OUTPUT

GL = boundary conditions on $\tilde{y}$ at the left-hand side of the problem interval

GR = boundary conditions on $\tilde{y}$ at the right-hand side of the problem interval

Description

BCAUX implements the boundary conditions (4.9)-(4.14) according to the scheme detailed in Table I. The λ's are chosen to be arbitrarily small ($\lambda_{1,2} = 10^{-4}$); their exact values do not affect the solution. The components of the unknown parameter vector $\tilde{p} = (p_1, p_2, p_3, p_4, p_5, p_6)$ are divided into two types. The set (p_1, p_3, p_5) clearly represents the solution at x = 0; it requires fairly reasonable initial guesses in order to reach a solution. The set (p_2, p_4, p_6), on the other hand, is trivially adjusted to match $\tilde{y}(1;\tilde{p})$, since x = 1 is chosen as the matching point. Therefore, its initial values are not critical.

The particular form chosen for r_2 and r_2' reflects a desire to keep all the components of $\tilde{p}$ roughly the same order of magnitude, i.e., $|p_i| \simeq 1$.

5.7. Subroutine RAAUX

Use

```
SUBROUTINE RAAUX(XL,XR,RMATCH,PARAM)
DIMENSION PARAM(6)
```

where

INPUT

PARAM = parameter vector $\tilde{p}$ supplied by D02AGF

OUTPUT

XL = left-hand boundary of the problem interval

XR = right-hand boundary of the problem interval

RMATCH = matching coordinate

Description

XL = XL0, initialized in MHDEQ as

XL0 = min [0.03, RHO(2)/A0]

and passed through COMMON/LOCMEQ/

XR = 1.0

RMATCH = 1.0

5.8. Subroutine PRSOL

Use

```
SUBROUTINE PRSOL(PARAM,RES,N1,ERR)
DIMENSION PARAM(N1),ERR(N1)
```

where

INPUT

PARAM = parameter vector $\tilde{p}$

N1 = dimension of $\tilde{y}$

ERR = error vector $\tilde{d}$ (4.21)

RES = sum of squares of ERR

Description

This is a diagnostic routine called by D02AGF just after integrating the $\tilde{y}[x; \tilde{p}^{(k)}]$ from XL to XR and just before reevaluating the current Jacobian. It provides diagnostic output to monitor the progress of the solution algorithm.

5.9. Subroutine AUX

Use

```
SUBROUTINE AUX(YP,Y,X,PARAM)
DIMENSION YP(6),Y(6),PARAM(6)
```

where

INPUT

Y = solution vector $\tilde{y}(x; p)$

X = the point $x \in [\delta_0, 1]$ at which $\tilde{y}'(x; \tilde{p})$ is required

PARAM = parameter vector $\tilde{p}$

OUTPUT

YP = $\tilde{y}'(x; \tilde{p})$

Description

AUX solves the system (4.15) by solving the equivalent first-order system

$$\underline{\underline{A}}\,\underline{y}' = \underline{b} \ . \tag{5.6}$$

where

$$\underline{y} = (r_0, r_0', E, E', r_2, r_2') \ . \tag{5.7}$$

The required values of pressure and current, $p'(x)$ and $I(x)$, $I'(x)$, are obtained from user-provided functions PRSFUN and CURFUN, respectively. The dimension of $\underline{\underline{A}}$ depends on the value of KMHD. For KMHD = 1 (solve for r_0 only) and KMHD = 2 (solve for r_0 and E only), the exempt variables are integrated from separate quadratic driving equations that are independent of the Newton iteration loop, since they are not functions of the parameter vector $\underline{p}$. For KMHD = 3, the full 3-equation system is solved.

A simple Kramer's rule algorithm is in place to solve the linear system when KMHD = 2 or 3. If the system is singular, an error flag is set and a return is made to the calling program leaving the last solution intact.

5.10. Subroutine SDOT

Use

```
FUNCTION SDOT(N,X,IX,Y,IY)
DIMENSION X(N),Y(N)
```

where

INPUT

N = number of elements in the summation

X = vector x

IX = array increment in x

Y = vector y

IY = array increment in y

OUTPUT

SDOT = scalar product of the vectors x and y, with adjacent elements IX and IY apart, respectively

Description

SDOT performs the scalar inner product on two vectors $\underline{x}$ and $\underline{y}$:

$$SDOT = \sum_{i=1}^{n} x_i y_i = \underline{x} \cdot \underline{y} \ . \tag{5.8}$$

In the routine supplied, the algorithm is a simple FORTRAN program. It is preferable, if possible, to substitute an optimized machine-dependent routine to speed up the calculation. Up to 50% of the CPU time in VMOMS is spent in SDOT. An optimized assembly language routine, or the use of special software that uses a host computer's special processing capabilities, for instance, will result in substantial savings in execution time.

SDOT is used exclusively to calculate integrals of the type

$$\frac{1}{2\pi} \int_0^{2\pi} f(\theta) d\theta \tag{5.9}$$

using Simpson's rule

$$\frac{1}{2\pi}\int_0^{2\pi} f(\theta)d\theta = \int_0^1 f(x)dx = \sum_{i=1}^{NTHETA} WFCN(i)*F(i) \ , \qquad (5.10)$$

where WFCN is a weighting array defined by

WFCN(1) = 1.

WFCN(i) = 4. for i even

WFCN(i) = 2. for i odd

WFCN(NTHETA) = 1.

and

NTHETA = odd integer.

NTHETA is entered through NAMELIST /MOMIN/, and WFCN is initialized in MHDEQ.

5.11. Subroutine SINCOS

Use

SUBROUTINE SINCOS

Description

SINCOS sets up the sin(nθ) and cos(nθ) (with n = 1,2) arrays (S1TH, S2TH, C1TH, and C2TH) for the subroutine AUX.

5.12. Cubic spline interpolation subroutines SPLAAN, SEVAL, and SPEVAL

(1). SPLAAN

Use

```
SUBROUTINE SPLAAN(N,X,Y,B,C,D)
DIMENSION X(N),Y(N),B(N),C(N),D(N)
```

where

INPUT

N = number of data points (N ≥ 2)

X = abscissa array in strictly increasing order

Y = ordinate array

OUTPUT

B,C,D = spline coefficient arrays

Description

SPLAAN is a modification of the subroutine SPLINE, which is described fully in Ref. [7]. The latter constructs a piecewise cubic representation $[S(x_i)]$ for y(x) between data points by imposing the constraints that S, S', and S" be continuous at each x_i. Additionally, S" is specified at each of the end points to correspond exactly to the corresponding derivative of the end cubic that passes exactly through the four end data points on each side. SPLAAN replaces this last condition on the left-hand end point with the derivative condition $S'(x_1) = 0$. This is useful for representing the moment functions which are analytically required to satisfy the condition $r_n'(0) = 0$.

(2). $\begin{Bmatrix} \text{SEVAL} \\ \text{SPEVAL} \end{Bmatrix}$

Use

FUNCTION $\begin{Bmatrix} \text{SEVAL} \\ \text{SPEVAL} \end{Bmatrix}$(N,U,X,Y,B,C,D)

DIMENSION X(N),Y(N),B(N),C(N),D(N)

where

INPUT

N = number of data points (as in SPLAAN)

U = the abscissa at which the spline is to be evaluated

X = abscissa array (as in SPLINE)

Y = ordinate array (as in SPLINE)

B,C,D = spline coefficient arrays (as in SPLINE)

OUTPUT

{SEVAL / SPEVAL} = cubic spline {interpolant / derivative} at x

Description

The operation of SEVAL is described fully in [7]. SPEVAL is a modification of SEVAL in that it provides the derivative of the cubic spline interpolant.

5.13. D02AGF [4]

Use

```
SUBROUTINE D02AGF(HH,ERROR,PARERR,PARAM,C,N,N1,M1,AUX,BCAUX,RAAUX,
PRSOL,MAT,COPY,WSPACE,WSPAC1,WSPAC2,IFAIL)

DIMENSION ERROR(N),PARERR(N1),PARAM(N1),C(M1,N),COPY(N1,N1),
     WSPACE(N,9),WSPAC1(N),WSPAC2(N),MAT(N1,N1)
EXTERNAL AUX,BCAUX,RAAUX,PRSOL
```

where

INPUT

HH = initial step length used in the integration routine in D02AGF (initialized to 0.01 in MHDEQ).

ERROR = an error estimate array used to

(1) test the local error in the solution vector $\underline{y}$ during integration

(2) test the convergence of the difference $\underline{d}$ at the matching point x_m

PARERR = an error estimate array used to

(1) test the convergence of the kth iterate $\underline{p}^{(k)}$ in the Newton iteration

(2) to estimate the perturbation on the parameter vector $\underline{p}$ in computing the Jacobian

PARAM = initial estimate of the parameter vector $\underline{\tilde{p}}$

N = number of differential equations

N1 = number of parameters

M1 = number of points in internal grid

AUX,BCAUX, RAAUX,PRSOL, = subroutines, described elsewhere, declared EXTERNAL in MHDEQ

MAT,COPY, WSPACE,WSPAC1, WSPAC2 = work arrays

IFAIL = 1 (error flag initialization)

OUTPUT

HH = final step size used by integration routine

PARAM = solution parameter vector $\underline{\tilde{p}}$

C = the solution vector $\underline{\tilde{y}}$ at (M1) equidistant points

IFAIL = error flag (cf. Table III)

Description

D02AGF [3], as used here, is a suite of programs from the NAG program library [4]. Full documentation can be found there. The source code, with comments, is supplied with VMOMS for convenience. To facilitate an understanding of program flow, we have inserted an internal logical variable, VRBOSE, which acts as a flag to turn on or off diagnostic output (unit 13), depending on its data loaded value of .TRUE. or .FALSE., respectively.

Since this code was written to be easily used on machines other than the PDP-10, it was thought wise to substitute for those routines in D02AGF that generate machine-specific parameters, such as machine precision, etc. To this end, we have included the routines SPMPAR [8] and I1MACH [9]; these routines provide machine-dependent parameters for over 14 different computers.

5.14. Program PLOTCM

Description

PLOTCM is an independent program provided for the convenience of the user in postprocessing the plot data file generated by VMOMS on unit 12. It is written specifically to be used on the PDP-10 with DISSPLA graphics [10]. It is anticipated that its use for those outside the Oak Ridge National Laboratory Fusion Energy Division will be primarily as a guide to the contents of the plot file and as a guide for writing a plotting routine specific to the users own computing system. The output plots are described in the section "Description of Output."

6. Timing considerations

The CPU time for the calculation (MHDEQ) varies proportionately with n to n^2 (where n = KMHDEQ), depending on the values of the variables mentioned below as well as on the initial guess for $\tilde{p}$. Typical times on the DEC PDP-10 (KL10) for KMHDEQ = 1, 2, and 3 are shown in Table IV, along with the NAMELIST parameters used in these calculations.

The effects of various problem parameters on the timing are discussed briefly below. All times cited are for operation on the PDP-10.

6.1. ERROR/PARERR

The tighter the convergence criteria (ERROR and PARERR) are made, the longer the time required for a solution. For instance, for the case documented in Table IV, changing ERROR to 0.01 and PARERR to 0.001 results in a doubling of the execution time.

6.2. XLO

The critical singularity at $x = 0$ has a strong effect on the convergence rate of the Newton iteration, depending on its proximity to the starting point $x = XLO$ chosen by the user. In the case cited above, reducing YLO from 0.03 to 0.01 increases the execution time by nearly a factor of 2.

6.3. Internal grid size

The integration of the ODE's uses a variable step size fourth-order Runge-Kutta-Merson technique [11], and the solution vector $\tilde{y}(x;\tilde{p})$ is <u>not</u> stored at the internal x_i used in the integration. After the parameter vectors $\tilde{p}$ are obtained, the $\tilde{y}$'s are integrated one last time and evaluated on an internal, uniformly spaced grid over [XLO,1] specified by the user (cf. M1 in Sect. 5.13). In the case cited above, for a ten-point grid, the overhead for this final evaluation is 0.5 s. Because the vectors $\tilde{y}(x;\tilde{p})$ are smooth, we choose a coarse grid to

minimize the execution time and interpolate back onto the (finer) driver grid using cubic splines. Clearly, other options are available to the user if more resolution is required.

7. Test case

7.1. Parameters

The test case parameters are the same as those used in the timing test (cf. Table IV), with KMHD = 3.

7.2. Description of output

Portions of the test case output are included at the end of this report. The first page displays both the input parameters, as provided through NAMELIST/MOMIN/, and the output parameters that characterize the solution.

The second page provides a tabulation of the driving functions $p(\rho)$ and $I(\rho)$, as well as of the final solutions $[r_0(\rho),\ E(\rho),\ r_2(\rho)]$ and their first and second derivatives. These are all "internal" values, defined on the internal "reduced" grid XP. Other arrays appearing there are defined in Table V.

Some of the plots generated by PLOTCM are shown on the third page and display various 1- and 2-D results.

Acknowledgment

The authors would like to thank P. W. Gaffney for very useful discussions on numerical methods during the development of this work. This research was sponsored by the Office of Fusion Energy, U.S. Department of Energy, under contract W-7405-eng-26 with the Union Carbide Corporation.

References

1. L. L. Lao, S. P. Hirshman, and R. M. Wieland, Phys. Fluids 24 (1981) 1431.

2. H. K. Meier (private communication); H. K. Meier, S. P. Hirshman, D. J. Sigmar, and L. L. Lao, ORNL/TM-7584, Oak Ridge, Tennessee (1981).

3. I. Gladwell, Num. Anal. Rep. No. 35, Dept. of Math., University of Manchester, England (1979); "The Development of the Boundary Value Codes in the Ordinary Differential Equations Chapter of the NAG Library," in Codes for Boundary Value Problems in Ordinary Differential Equations, Ed. by B. Childs, M. Scott, J. W. Daniel, E. Denman, and P. Nelson, Springer-Verlag Lecture Notes in Computer Science, Vol. 76 (1979).

4. Numerical Algorithms Group (NAG) Library Manual, Downer's Grove, IL., Mark 8 (1981).

5. J. Walsh, "Boundary Value Problems in Ordinary Differential Equations," in The State of the Art in Numerical Analysis, Ed. by D. A. H. Jacobs, Academic Press, New York (1977).

6. W. A. Houlberg, S. E. Attenberger, and L. L. Lao, Proc. Int. Top. Mtg. on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems (1981) 426-445.

7. G. E. Forsythe, M. A. Malcolm, and C. B. Moler, Computer Methods for Mathematical Computations, Prentice-Hall, Englewood Cliffs, N.J. (1977).

8. B. S. Garbow, K. E. Hillstrom, and J. M. More, Argonne National Laboratory Report ANL-80-68 (July 1980).

9. P. A. Fox, A. D. Hall, and N. L. Schryer, ACM Transactions on Mathematical Software, Vol. 4, No. 2 (June 1978) 177-188.

10. DISSPLA, ISSCO, San Diego, California 92121.

11. R. H. Merson, Proc. Symp. Data Processing, Weapons Research Establishment, Salisbury, S. Australia (1957); J. D. Lambert, Computational Methods in Ordinary Differential Equations, Wiley, New York (1977).

TABLE I. Boundary Conditions

Moment	$\tilde{y}(\delta_o)$	$\tilde{y}(1)$
$\tilde{y}_1$: r_0	$p_1 - \lambda_1\delta_0^2$	r_c
$\tilde{y}_2$: r_0'	$-2\lambda_1\delta_0^2$	p_2
$\tilde{y}_3$: E	$p_3 + \lambda_2\delta_0^2$	E_a
$\tilde{y}_4$: E'	$2\lambda_2\delta_0$	p_4
$\tilde{y}_5$: r_2	$10p_5\delta_0^4$	D_a
$\tilde{y}_6$: r_2'	$40p_5\delta_0^3$	p_6

Constants

$\lambda_1 = 10^{-4}$

$\lambda_2 = 10^{-4}$

Initial iterate

$\tilde{p}_1(0) = R_a/a + 0.1$

$\tilde{p}_2(0) = -0.4$

$\tilde{p}_3(0) = 1.0 + (E_a - 1.0)/2.0$

$\tilde{p}_4(0) = 0.$

$\tilde{p}_5(0) = 1.0$

$\tilde{p}_6(0) = 2.0 * D_a$

TABLE II. NAMELIST/MOMIN/

Variable (type)	Default value	Description
KMHD (I)	3	Dimension of moments solution (KMHDEQ = 1,2, or 3)
IPARAM (I)	0	Parameter initialization switch for shooting code
RO (R)	94.0	Major plasma radius (cm)
AO (R)	26.0	Minor plasma radius (cm)
BTO (R)	9100.0	Toroidal magnetic field at RO (G)
PO (R)	1.32×10^{17}	Plasma pressure on magnetic axis (eV/cm^3)
PA (R)	2.0	Pressure profile parameter
PB (R)	1.0	Pressure profile parameter
AIO (R)	200×10^3	Toroidal current (A)
AIA (R)	2.0	Current density parameter
XEO (R)	1.55	Plasma elongation on boundary
XDO (R)	0.18	Plasma triangularity on boundary
XE1 (R)	1.0	Plasma elongation at magnetic axis (KMHDEQ = 1, only)

TABLE III. Error Flags

IFAIL	ERROR
1	N1 > N in D02AGF
2	Integration failed to converge while updating the Jacobian matrix
3	$x_m \neq$ [XLO,1]
4	Integration failed to converge while shooting to x = 1
5	The Jacobian matrix is singular
6	The parameter vector update is unsatisfactory
7	The Newton iteration failed
8	The linear system in AUX is singular

TABLE IV. Timing Comparisons

Namelist values

RO = 94.0	AO = 26.0	BTO = 9.1×10^3	AIOAM = $200. \times 10^3$
PO = 1.32×10^{17}	PA = 2.0	PB = 1.0	
AIA = 2.0	XE0 = 1.55	XE1 = 1.1	XD0 = 0.18

D02AGF parameters

HH = 0.01 XL0 = 0.03 ERROR = 6*5. $\times 10^{-2}$ PARERR = 6*1. $\times 10^{-2}$

KMHD	Execution Time (s) on PDP-10
3	3.2
2	2.3
1	1.4

TABLE V. OUTPUT Description

Name	Definition	Description
Q	$q(x)$	Safety factor
JFLUX	$\tilde{j}(x)$	Equivalent radial current density (stat-amps)
PSI	$\frac{\chi(0) - \chi(x)}{\chi(0) - \chi(a)}$	Poloidal flux function
F	$F(x)/F_0$	Poloidal current
FFP	$\frac{F(x)F'(x)}{F_0^2}$	
JMDPL(IN)	$j(x, \theta = 0^\circ)$	Inboard midplane current (stat-amps)
JMDPL(OUT)	$j(x, \theta = 180^\circ)$	Outboard midplane current (stat-amps)
GSQRT	$\langle\sqrt{g}\rangle$	Metric elements (dimensionless)
GTTGS	$\langle g_{\theta\theta}/\sqrt{g}\rangle$	
GSGPP	$\langle\sqrt{g}/g_{\phi\phi}\rangle$	
GRHO	$\langle\sqrt{g}\ \lvert\nabla\rho\rvert^2\rangle/\langle\sqrt{g}\rangle$	
ARIF	$\langle\sqrt{g}/R\rangle/\langle\sqrt{g}\rangle$	
AR2IF	$\langle\sqrt{g}/R^2\rangle/\langle\sqrt{g}\rangle$	

TABLE VI. Subroutine Index

Program	Operations
MDRIV	Sets up input profiles for MHDEQ; does I/O for results
MHDEQ	Initializes parameters and arrays required internally in AUX; interpolates D02AGF solution back onto driver grid
GEOTRN	Converts between the geometrical boundary conditions of the external code and the metric boundary conditions corresponding to the internal geometry
D02AGF	A NAG [4] routine for solving a system of nonlinear coupled ordinary differential equations (boundary condition problem) using the shooting method and a modified Newton iteration
AUX	For a given x, $\underline{\tilde{y}}(x)$ and $\underline{\tilde{y}}'(x)$, constructs a $y''(x)$
BCAUX	Constructs the boundary conditions as a function of the unknown parameters $\underline{p}$
RAAUX	Defines the endpoints of the integration interval as well as the matching point x_m
PRSOL	A diagnostic routine that prints out algorithmic convergence information during the course of the Newton iteration
SDOT	An inner scaler product routine
SPLAAN SEVAL SPEVAL	A suite of codes for cubic spline interpolation
SINCOS	An initialization routine to evaluate sin nθ, cos nθ, n = 1,2
GPASMA	Given the moment solution $\{r_0, E, r_2\}$ generates metrics as well as other quantities of interest $(\bar{\psi}, j, F)$ on the internal grid
PRTMEQ	Prints out moment solution vectors as well as quantities calculated in GPASMA
PLTMEQ	Generates a data file for subsequent postprocessing by user-provided plotting routine

TABLE VII. I/O Units Used by VMOMS

Unit Number	Purpose
INPUT	
10 (NIN)	NAMELIST input
OUTPUT	
5 (NTTY)	Terminal output
11 (NOUT)	Output file
12 (NPLOT)	Data file for input to plotting program (PLOTCM)
13 (NDEBUG)	D02AGF debugging output

TABLE VIII. Common Blocks

Name	Used By	Contents
CD02	GPASMA MHDEQ PLTMEQ PRTMEQ	Solution vectors $\tilde{y}$ and $\tilde{p}$
CFEVAL	AUX D02AGF	Function evaluation counter for D02AGF debug option
CINIT	CURFUN MHDEQ PRSFUN PRSOL	Initialization flags
LOCD02	MHDEQ PRTMEQ	Error arrays, work space for D02GAF
LOCGPA	GPASMA PLTMEQ PRTMEQ	Derived plasma scalar parameters
LOCMEQ	AUX BCAUX GPASMA MHDEQ PLTMEQ PRTMEQ RAAUX	Internal scalars used to construct $\tilde{y}''$
NIO	AUX D02AGF MDRIV PLTMEQ PRSOL PRTMEQ P01AAF	Input/output unit numbers
SCFUNC	AUX GPASMA MHDEQ SINCOS	the sin $n\theta$, cos $n\theta$ arrays
SWORK1	CURFUN MDRIV PRSFUN PLTMEQ PRTMEQ	Pressure and current profile arrays

TABLE VIII. Common Blocks (continued)

Name	Used By	Contents
SWORK2	AUX GPASMA MHDEQ	Internal θ-arrays used in constructing flux surface averages
SWORK3	GPASMA PLTMEQ PRTMEQ	Derived plasma profiles (GPASMA)
VINPUT	MDRIV PRTMEQ	Input parameters
VOUPUT	MDRIV PLTMEQ PRTMEQ	Output arrays
WEIGHT	AUX GPASMA MHDEQ	Weight function array for scalar product

Figure Captions

Fig. 1. Definition of (ρ,θ) flux surface geometry.

Fig. 2. Definition of parameters used in transformation of elongation and triangularity from geometric units to moments units.

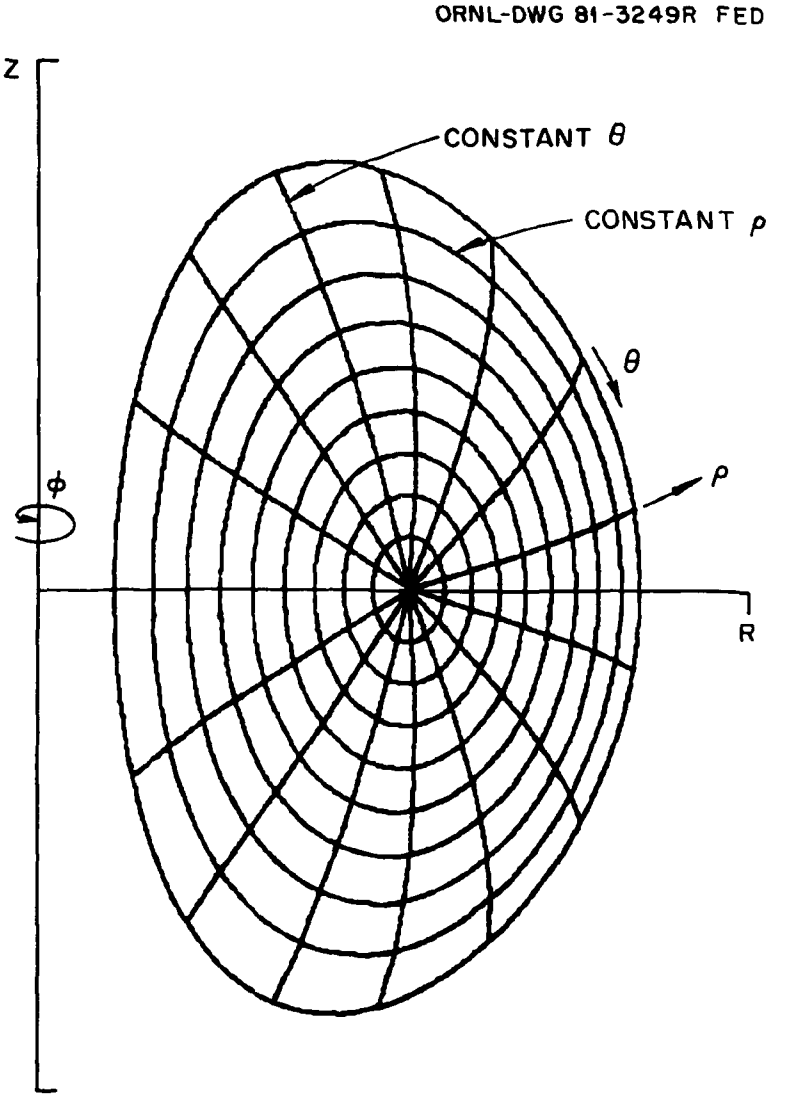


Fig. 1

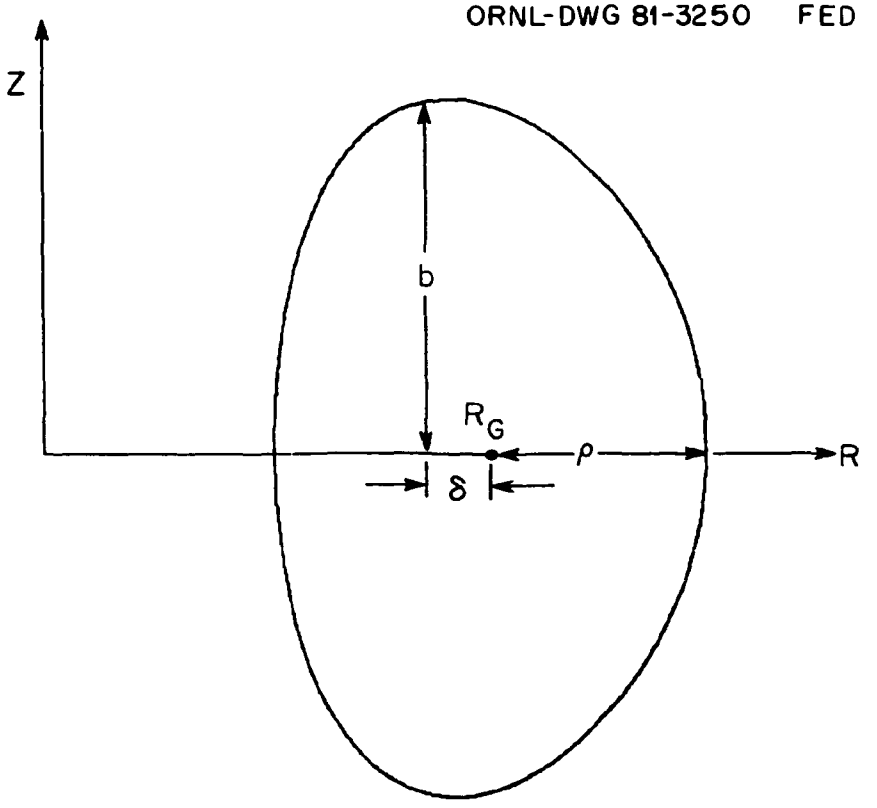


Fig. 2

```
****************** PLASMA  PARAMETERS **********************

          RO =  94.0          CM
          AO =  26.0          CM
          XE0 =  1.55         GEOMETRIC
          XE1 =  1.10         GEOMETRIC
          XD0 = 0.18          GEOMETRIC
          BTO =  9100.0       GAUSS
          PO =  1.32E+17      EV/CM**3
          PA = 2.00
          PB = 1.00
          AIOAM = 2.00E+05    AMPS
          AIA = 2.00


****************** D02AGF  PARAMETERS ******************

          KMHD   =    3
          HHO =  .01000
          XLO =  .03000
          PARERR =   1.00E-03  1.00E-03  1.00E-03  1.00E-03  1.00E-03  1.00E-03
          ERROR  =   1.00E-02  1.00E-02  1.00E-02  1.00E-02  1.00E-02  1.00E-02


****************** INTERNAL  PARAMETERS ******************

          BETAPO =   2.25
          FO =   4.28E+06 AMPS
          QO =   1.64E+00
          PSICO =   1.70E+07 GAUSS-CM**2
          E0 = 1.54           MOMENTS
          E1 = 1.10           MOMENTS
          DO =  .046          MOMENTS
```

Sample Output

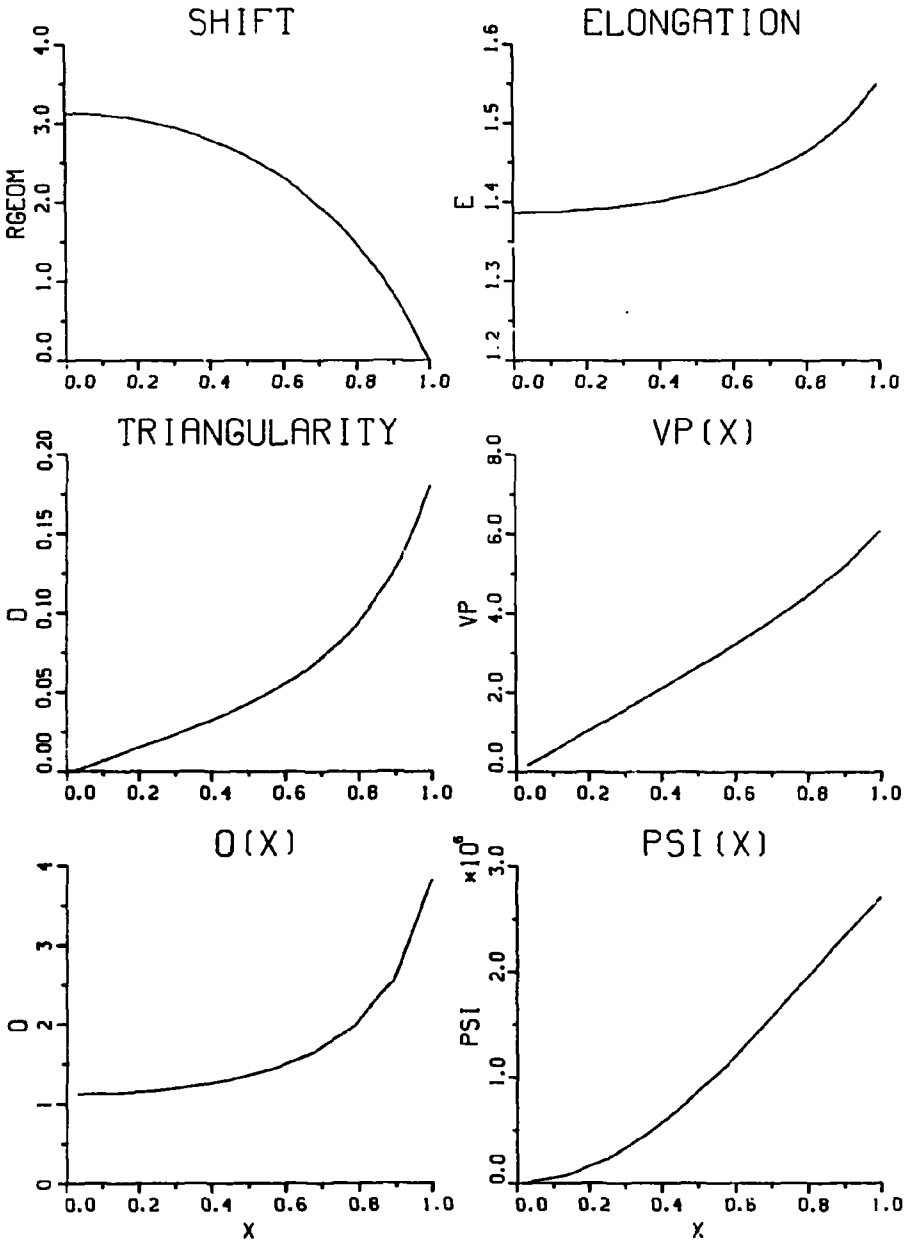
********** MOMENTS(INTERNAL) **********

I	X	P-HAT	PP-HAT	I-HAT	IP-HAT
1	0.03	9.991E-01	-6.000E-02	1.799E-03	1.199E-01
2	0.14	9.810E-01	-2.756E-01	3.761E-02	5.407E-01
3	0.25	9.397E-01	-4.911E-01	1.170E-01	9.230E-01
4	0.35	8.752E-01	-7.067E-01	2.341E-01	1.237E+00
5	0.46	7.874E-01	-9.222E-01	3.800E-01	1.452E+00
6	0.57	6.764E-01	-1.138E+00	5.425E-01	1.539E+00
7	0.68	5.421E-01	-1.353E+00	7.061E-01	1.467E+00
8	0.78	3.846E-01	-1.569E+00	8.520E-01	1.207E+00
9	0.89	2.039E-01	-1.784E+00	9.584E-01	7.278E-01
10	1.00	0.000E+00	-2.000E+00	1.000E+00	3.943E-04

I	X	R0	R0P	R0PP	E0	E0P	E0PP	D0	D0P	D0PP
1	0.03	3.736E+00	-6.000E-06	-7.439E-01	1.386E+00	6.000E-06	1.354E+00	9.219E-06	1.229E-03	7.643E-02
2	0.14	3.734E+00	-2.596E-02	-1.961E-01	1.388E+00	2.346E-02	1.714E-01	3.538E-04	5.217E-03	3.874E-02
3	0.25	3.730E+00	-4.798E-02	-2.155E-01	1.391E+00	4.330E-02	1.958E-01	1.151E-03	9.692E-03	4.452E-02
4	0.35	3.724E+00	-7.300E-02	-2.521E-01	1.397E+00	6.615E-02	2.316E-01	2.471E-03	1.500E-02	5.492E-02
5	0.46	3.714E+00	-1.032E-01	-3.129E-01	1.406E+00	9.405E-02	2.910E-01	4.438E-03	2.183E-02	7.361E-02
6	0.57	3.701E+00	-1.418E-01	-4.132E-01	1.418E+00	1.303E-01	3.894E-01	7.274E-03	3.143E-02	1.083E-01
7	0.68	3.683E+00	-1.947E-01	-5.842E-01	1.434E+00	1.806E-01	5.610E-01	1.140E-02	4.637E-02	1.770E-01
8	0.78	3.658E+00	-2.727E-01	-8.947E-01	1.458E+00	2.568E-01	8.946E-01	1.766E-02	7.237E-02	3.255E-01
9	0.89	3.623E+00	-3.984E-01	-1.515E+00	1.492E+00	3.886E-01	1.674E+00	2.789E-02	1.238E-01	6.827E-01
10	1.00	3.570E+00	-6.244E-01	-2.869E+00	1.544E+00	6.510E-01	3.759E+00	4.555E-02	2.358E-01	1.566E+00

I	X	Q	JFLUX (STATAMPS)	PSI	F	FFP	JMDPL(IN) (STATAMPS)	JMDPL(OUT) (STATAMPS)
1	0.03	1.121E+00	4.072E+11	4.403E-03	1.001E+00	-2.950E-03	4.078E+11	4.066E+11
2	0.14	1.135E+00	3.990E+11	2.883E-02	9.998E-01	-1.286E-02	4.014E+11	3.968E+11
3	0.25	1.168E+00	3.802E+11	8.377E-02	9.980E-01	-2.018E-02	3.829E+11	3.782E+11
4	0.35	1.226E+00	3.511E+11	1.672E-01	9.957E-01	-2.321E-02	3.514E+11	3.523E+11
5	0.46	1.314E+00	3.118E+11	2.758E-01	9.933E-01	-2.074E-02	3.056E+11	3.208E+11
6	0.57	1.446E+00	2.629E+11	4.054E-01	9.915E-01	-1.233E-02	2.448E+11	2.859E+11
7	0.68	1.648E+00	2.052E+11	5.503E-01	9.909E-01	1.560E-03	1.683E+11	2.508E+11
8	0.78	1.972E+00	1.399E+11	7.040E-01	9.921E-01	1.929E-02	7.654E+10	2.204E+11
9	0.89	2.549E+00	6.971E+10	8.578E-01	9.951E-01	3.779E-02	-2.853E+10	2.046E+11
10	1.00	3.816E+00	3.037E+07	1.000E+00	1.000E+00	5.201E-02	-1.417E+11	2.293E+11

I	X	GSQRT	GTTGS	GSQPP	GRHO	AR1F	AR2IF
1	0.03	1.554E-01	8.463E-03	1.113E-02	7.602E-01	2.677E-01	7.166E-02
2	0.14	7.144E-01	3.888E-02	5.132E-02	7.590E-01	2.679E-01	7.184E-02
3	0.25	1.277E+00	6.938E-02	9.228E-02	7.565E-01	2.685E-01	7.226E-02
4	0.35	1.847E+00	1.000E-01	1.347E-01	7.524E-01	2.695E-01	7.295E-02
5	0.46	2.428E+00	1.308E-01	1.796E-01	7.466E-01	2.710E-01	7.398E-02
6	0.57	3.028E+00	1.619E-01	2.285E-01	7.391E-01	2.731E-01	7.546E-02
7	0.68	3.657E+00	1.935E-01	2.838E-01	7.303E-01	2.763E-01	7.760E-02
8	0.78	4.336E+00	2.261E-01	3.502E-01	7.216E-01	2.811E-01	8.077E-02
9	0.89	5.107E+00	2.620E-01	4.381E-01	7.177E-01	2.888E-01	8.578E-02
10	1.00	6.070E+00	3.114E-01	5.730E-01	7.413E-01	3.023E-01	9.441E-02


SHIFT
RGEOM
0.0
1.0
2.0
3.0
4.0
0.0 0.2 0.4 0.6 0.8 1.0
ELONGATION
E
1.2
1.3
1.4
1.5
1.6
0.0 0.2 0.4 0.6 0.8 1.0
TRIANGULARITY
D
0.00
0.05
0.10
0.15
0.20
0.0 0.2 0.4 0.6 0.8 1.0
VP(X)
VP
0.0
2.0
4.0
6.0
8.0
0.0 0.2 0.4 0.6 0.8 1.0
Q(X)
Q
0
1
2
3
4
0.0 0.2 0.4 0.6 0.8 1.0
X
PSI(X)
×10⁶
PSI
0.0
1.0
2.0
3.0
0.0 0.2 0.4 0.6 0.8 1.0
X