NRL Report 7706

UNCLASSIFIED

PROP-I: An Efficient Implicit Algorithm for Calculating Nonlinear Scalar Wave Propagation in the Fresnel Approximation

PETER B. ULRICH

Optical Radiation Branch
Optical Sciences Division

May 29, 1974

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

UNCLASSIFIED

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 7706	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* PROP-I: AN EFFICIENT IMPLICIT ALGORITHM FOR CALCULATING NONLINEAR SCALAR WAVE PROPAGATION IN THE FRESNEL APPROXIMATION		5. TYPE OF REPORT & PERIOD COVERED Final report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* Peter B. Ulrich		8. CONTRACT OR GRANT NUMBER*(s)*
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem R05-31.304 U1754
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Navy Naval Ordnance Systems Command Washington, D.C. 20360		12. REPORT DATE May 29, 1974 13. NUMBER OF PAGES 68
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Laser beams	Light transmission
Propagation	Thermal blooming
Atmosphere	Numerical methods

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

An algorithm has been developed to advance rapidly the solution of the parabolic equation which describes the problem of propagation of intense laser beams in absorbing media. The method uses an alternating-direction implicit (ADI) technique together with either central differencing or a Galerkin method with linear spline basis to represent the transverse Laplacian in the partial differential equation. A guide to the code, a listing of the Fortran program, and a flowchart are provided.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

UNCLASSIFIED

CONTENTS

1. INTRODUCTION 1

2. TRANSFORMATIONS OF THE PARTIAL DIFFERENTIAL EQUATION 1

3. IMPLICIT ALGORITHM FOR *Z*-INTEGRATION 3

4. TRANSVERSE DISCRETIZATION 5

4.1 Central Difference Approximation (CDA) 5
4.2 Galerkin Method 6

5. BOUNDARY VALUES 9

6. DESCRIPTION OF PROP-I 10

6.1 Common Arrays 10
6.2. Common Variables and Constants 11
6.3. Subroutines 11

7. CONCLUSIONS 13

ACKNOWLEDGMENT 13

REFERENCES 14

APPENDIX A — Fortran Listings 15

UNCLASSIFIED

PROP-I: AN EFFICIENT IMPLICIT ALGORITHM FOR CALCULATING NONLINEAR SCALAR WAVE PROPAGATION IN THE FRESNEL APPROXIMATION

1. INTRODUCTION

This report develops in detail an algorithm for rapidly advancing the solution of the parabolic equation that describes the propagation of an intense laser beam in an absorbing medium. The heating of the medium by laser radiation causes the index of refraction to change; this in turn modifies the propagation of the beam. A steady wind, with or without beam-sluing, is assumed to sweep out the heated air, leading to a steady state. The relevant equations were derived in a previous report (1) in which it was shown that scalar wave theory in the Fresnel approximation, together with linearized hydrodynamics, provide very good approximations for the parameters of interest to the Navy and lead to the parabolic partial differential equation for the complex amplitude. It is assumed that the reader of this report is familiar with Secs. I and II of the earlier report, which describes the physics of the problem.

Section 2 of this report gives a new set of transformations of the partial differential equation. This set is similar to the set employed successfully in a pulsed-beam computer program (2). In Sec. 3 the implicit algorithm that replaces the explicit algorithm of Ref. 1 is derived. Section 4 is devoted to two distinct discretization schemes for the coordinates transverse to the propagation direction. Section 5 discusses boundary values. Section 6 is a "handbook" for use of the program. A flowchart and Fortran listings are presented in Appendix A.

2. TRANSFORMATIONS OF THE PARTIAL DIFFERENTIAL EQUATION

The starting point is Eq. (1) of Ref. 1:

$$2ik \frac{\partial \psi_0}{\partial z} + \nabla_\perp^2 \psi_0 + k^2(n^2 - 1)\psi_0 = 0 \tag{1}$$

where

ψ_0 is the scalar wave amplitude of the light beam, $k = 2\pi/\lambda$, (λ = wavelength of the light beam)

z is the propagation direction

$\nabla_\perp^2 \equiv \partial^2/\partial x^2 + \partial^2/\partial y^2$ is the Laplacian in transverse coordinates

n is the index of refraction of the medium ($n = 1$ in vacuum).

Note: Manuscript submitted December 14, 1973.

Define $\bar{x} = x/a$, $\bar{y} = y/a$, and $\zeta = z/ka^2$, where a is a scale length associated with the transverse beam size at $z = 0$. Then, if $\psi_1(\bar{x}, \bar{y}, \zeta) = \psi_0(x, y, z)$, Eq. (1) becomes

$$2i \frac{\partial \psi_1}{\partial \zeta} + \bar{\nabla}_\perp^2 \psi_1 + k^2 a^2 (n^2 - 1) \psi_1 = 0 . \tag{2}$$

Introduce a phase and scale change to remove the oscillations and amplitude growth of a focused Gaussian beam in vacuum:

$$\psi_2(\bar{x}, \bar{y}, \zeta) = \sqrt{D(\zeta)} \exp\left[-i \frac{\bar{x}^2 + \bar{y}^2}{2} \frac{\partial}{\partial \zeta} \ln D^{1/2} - i \tan^{-1}\left(\frac{\zeta}{1 - \zeta/\hat{f}}\right)\right] \psi_1(\bar{x}, \bar{y}, \zeta)$$

where

$$D(\zeta) = \zeta^2 + (1 - \zeta/\hat{f})^2$$

$$\hat{f} = f/ka^2,$$

so that

$$2i \frac{\partial \psi_2}{\partial \zeta} + \bar{\nabla}_\perp^2 \psi_2 - \frac{\bar{x}^2 + \bar{y}^2}{D^2} \psi_2 + 2i \frac{\partial}{\partial \zeta} \ln D^{1/2} \left(\bar{x} \frac{\partial \psi_2}{\partial \bar{x}} + \bar{y} \frac{\partial \psi_2}{\partial \bar{y}} \right) + \frac{2}{D} \psi_2 + k^2 a^2 (n^2 - 1) \psi_2 = 0. \tag{3}$$

The transverse coordinates are made to follow the vacuum trajectories of the focused beam by the transformation

$$\tilde{x} = \bar{x}/\sqrt{D}, \quad \tilde{y} = \bar{y}/\sqrt{D}, \quad \psi_3(\tilde{x}, \tilde{y}, \zeta) = \psi_2(\bar{x}, \bar{y}, \zeta),$$

so that

$$2i \frac{\partial \psi_3}{\partial \zeta} + \frac{1}{D} \tilde{\nabla}_\perp^2 \psi_3 - \frac{\tilde{x}^2 + \tilde{y}^2}{D} \psi_3 + \frac{2}{D} \psi_3 + k^2 a^2 (n^2 - 1) \psi_3 = 0. \tag{4}$$

The factors of $1/D$ can be removed with the transformation $dZ/d\zeta = 1/D$, which for equal steps in the new variable Z, gives smaller steps near the focus for the variable ζ. Then, defining $\psi_4(\tilde{x}, \tilde{y}, Z) = \psi_3(\tilde{x}, \tilde{y}, \zeta)$, Eq. (4) becomes

$$2i \frac{\partial \psi_4}{\partial Z} + \tilde{\nabla}_\perp^2 \psi_4 - (\tilde{x}^2 + \tilde{y}^2 - 2) \psi_4 + D k^2 a^2 (n^2 - 1) \psi_4 = 0. \tag{5}$$

UNCLASSIFIED

3. IMPLICIT ALGORITHM FOR Z-INTEGRATION

Following Herrmann and Bradley (3), define

$$g(\tilde{x}, \tilde{y}, Z) = \tilde{x}^2 + \tilde{y}^2 - 2 - k^2 a^2 D(Z)(n^2 - 1)$$

and let

$$\psi_4(\tilde{x}, \tilde{y}, Z) = \Phi(\tilde{x}, \tilde{y}, Z) \exp\left[-\frac{i}{2} \int_{Z_0}^{Z} g(\tilde{x}, \tilde{y}, Z')\, dZ'\right]$$

where Z_0 is a constant. Let

$$\Gamma(\tilde{x}, \tilde{y}, Z) \equiv \frac{1}{2} \int_{Z_0}^{Z} g(\tilde{x}, \tilde{y}, Z')\, dZ'.$$

Then Eq. (5) becomes

$$2i \frac{\partial \Phi}{\partial Z} + H(Z)\Phi = 0 \tag{6}$$

where the operator H is defined to be

$$H(Z) \equiv e^{i\Gamma} \tilde{\nabla}_\perp^2 e^{-i\Gamma}.$$

The specification of $H(Z)$ for numerical work is now required. Equation (6) is a nonlinear partial differential equation which is first order in Z. A difference scheme solution will be built up from elementary steps that create new values of the dependent variable Φ on a plane Z_c = constant from known values on a plane $Z_c - \Delta Z$. If ΔZ is small enough (this will be made explicit below), it is virtually immaterial where $H(Z)$ is defined to operate in the interval $[Z_c - \Delta Z, Z_c]$. Various essentially equivalent schemes can be designed by specification of the plane where $H(Z)$ is chosen to operate (the different schemes may vary in accuracy, however). For reasons which become apparent below, $H(Z)$ is selected to operate *midway* between the two planes. Thus, wherever Z appears explicitly in $H(Z)$, it is replaced by $Z_c - \Delta Z/2$, so that H becomes

$$H = e^{i\Gamma(Z_c - \Delta Z/2)} \tilde{\nabla}_\perp^2 e^{-i\Gamma(Z_c - \Delta Z/2)}.$$

In addition, the lower limit Z_0 in $\Gamma(Z)$ is still free to be chosen. The choice $Z_0 = Z_c - \Delta Z/2$ is made as well. Then H becomes

$$H = \tilde{\nabla}_\perp^2,$$

so that Eq. (6) is now

$$2i \frac{\partial \Phi}{\partial Z} + \tilde{\nabla}_\perp^2 \Phi = 0. \tag{7}$$

Note that Z_0 is redefined at each step, ΔZ. Thus, the complete module for the iterative construction of a solution to the problem consists of three steps:

1. Given data $\psi_4(\tilde{x}, \tilde{y}, Z_1)$ on some plane Z_1 = constant, form the dependent variable $\Phi(\tilde{x}, \tilde{y}, Z_1)$ by the phase change

$$\Phi(\tilde{x}, \tilde{y}, Z_1) = \psi_4(\tilde{x}, \tilde{y}, Z_1) \exp\left[\frac{i}{2} \int_{Z_1+\Delta Z/2}^{Z_1} g(\tilde{x}, \tilde{y}, Z')\, dZ'\right].$$

2. Solve for $\Phi(\tilde{x}, \tilde{y}, Z_2)$ where $Z_2 = Z_1 + \Delta Z$, using a difference scheme for Eq. (7).

3. Recreate the dependent variable ψ_4, now at the new plane Z_2 = constant by the inverse phase transformation

$$\psi_4(\tilde{x}, \tilde{y}, Z_2) = \Phi(\tilde{x}, \tilde{y}, Z_2) \exp\left[-\frac{i}{2} \int_{Z_1+\Delta Z/2}^{Z_2} g(\tilde{x}, \tilde{y}, Z')\, dZ'\right].$$

This completes the elementary sequence. For implicit schemes, Eq. (7) is self-starting and needs only two planes of data, instead of three as required by the algorithm of Ref. 1. The value of $\Gamma(Z)$ is nonzero even in vacuum because of the $\tilde{x}^2 + \tilde{y}^2 - 2$ term in $g(\tilde{x}, \tilde{y}, Z)$. Above it was stated that ΔZ must be "small enough"; this translates into a restriction that the phase change $0.5 \int g dZ'$ per step ΔZ be small (for example, $\leqslant 0.5$ rad).

Since the operator H is separable as

$$H = H_x + H_y = \partial^2/\partial\tilde{x}^2 + \partial^2/\partial\tilde{y}^2,$$

the alternating-direction implicit (ADI) algorithm (4) can be used. The advancement in Z goes in pairs. Given $\Phi(Z_n) \equiv \Phi(n\Delta Z)$, $n = 0, 1, 2, \ldots, N_f$ (where $N_f \Delta Z$ is the range for which the solution is desired) the first step is explicit in y and implicit in x;

$$2i\, \frac{\Phi(Z_{n+1}) - \Phi(Z_n)}{\Delta Z} = -H_x \Phi(Z_{n+1}) - H_y \Phi(Z_n).$$

The second step is implicit in y and explicit in x;

$$2i\, \frac{\Phi(Z_{n+2}) - \Phi(Z_{n+1})}{\Delta Z} = -H_x \Phi(Z_{n+1}) - H_y \Phi(Z_{n+2}).$$

These two steps can be written as

$$\left(1 - i\, \frac{\Delta Z}{2} H_x\right) \Phi_{n+1} = \left(1 + i\, \frac{\Delta Z}{2} H_y\right) \Phi_n \tag{8}$$

and

$$\left(1 - i\, \frac{\Delta Z}{2} H_y\right) \Phi_{n+2} = \left(1 + i\, \frac{\Delta Z}{2} H_x\right) \Phi_{n+1} \tag{9}$$

UNCLASSIFIED

where Eq. (8) is completely solved before Eq. (9) and $\Phi_n \equiv \Phi(Z_n)$. This algorithm is attractive for two reasons: (a) It is absolutely stable (4) for any choice of step size ΔZ. (There are accuracy limitations (3) on ΔZ, however.) (b) The matrixes which are to be inverted at each step are band-diagonal for the conventional choices of transverse discretization (this will become evident below) and are thus rapidly inverted.

4. TRANSVERSE DISCRETIZATION

Two methods have been used to represent the transverse Laplacían in Eq. (7): (a) The central difference approximation, and (b) the Galerkin method (5) with a linear spline basis.

4.1. Central Difference Approximation (CDA)

This employs the conventional replacement of derivatives by differences (1), so that

$$H_x\Phi(\tilde{x},\tilde{y},Z) = \frac{\Phi(\tilde{x}+\Delta\tilde{x},\tilde{y},Z) - 2\Phi(\tilde{x},\tilde{y},Z) + \Phi(\tilde{x}-\Delta\tilde{x},\tilde{y},Z)}{(\Delta\tilde{x})^2}$$

and

$$H_y\Phi(\tilde{x},\tilde{y},Z) = \frac{\Phi(\tilde{x},\tilde{y}+\Delta\tilde{y},Z) - 2\Phi(\tilde{x},\tilde{y},Z) + \Phi(\tilde{x},\tilde{y}-\Delta\tilde{y},Z)}{(\Delta\tilde{y})^2}$$

With the introduction of the notation

$$\Phi(\ell\Delta\tilde{x}, m\Delta\tilde{y}, n\Delta Z) \equiv \phi^n_{\ell,m} ,$$

$$\eta_x = \frac{\Delta Z}{2(\Delta\tilde{x})^2}, \qquad \eta_y = \frac{\Delta Z}{2(\Delta\tilde{y})^2},$$

Equations (8) and (9) become

$$(1+2i\eta_x)\phi^{n+1}_{\ell,m} - i\eta_x\phi^{n+1}_{\ell+1,m} - i\eta_x\phi^{n+1}_{\ell-1,m} = (1-2i\eta_y)\phi^n_{\ell,m} + i\eta_y\phi^n_{\ell,m+1} + i\eta_y\phi^n_{\ell,m-1} \tag{10}$$

and

$$(1+2i\eta_y)\phi^{n+2}_{\ell,m} - i\eta_y\phi^{n+2}_{\ell,m+1} - i\eta_y\phi^{n+2}_{\ell,m-1} = (1-2i\eta_x)\phi^{n+1}_{\ell,m} + i\eta_x\phi^{n+1}_{\ell+1,m} + i\eta_x\phi^{n+1}_{\ell-1,m} . \tag{11}$$

Solution of these equations requires that the matrix of the coefficients of the dependent variables on the left-hand sides be inverted. The matrix is tridiagonal (nonzero entries only on the diagonal and above and below) so that the solution can be rapidly achieved by single-pass Gaussian elimination (6).

4.2 Galerkin Method

After discretization in Z, Eq. (7) is a set of partial differential equations in the two transverse variables. Write the set as $L(\Phi) = 0$, where L is a differential operator in two variables and Φ is a vector whose components are the transformed field amplitudes on the Z plane from which the solution is proceeding. An approximate solution in the form

$$\bar{\Phi}_i(x, y) = \sum_{i=1}^{m} C_i \Phi_i(x, y)$$

is sought, where $\Phi_i(x, y)$, $i = 1, \ldots, m$ is a system of linearly independent functions satisfying the same boundary conditions as $\Phi(x, y)$. The $\{\Phi_i\}$ represents a subset of a set of functions complete in the region D where a solution is sought. The coefficients C_i are determined by m conditions of orthogonality

$$\iint_D L(\bar{\Phi})\, \Phi_i \, dxdy = 0$$

or

$$\iint_D L\left(\sum_{j=1}^{m} C_j \Phi_j(x, y)\right) \Phi_i(x, y)\, dxdy = 0 .$$

Because of the separability of the operator $H(Z) = H_x + H_y$, the functions $\Phi_i(x, y)$ can be written as products of basis functions in each variable. Thus a decomposition of the form

$$\bar{\Phi}^n(x, y) = \sum_{i,j} S_i(x) S_j(y) C^n_{i,j} \tag{12}$$

is taken to approximate the fields at each plane $Z = n\Delta Z$. The basis functions $S_i(x)$ are the linear spline functions defined by

$$S_i(x) = 0, \quad x \leqslant x_{i-1}, \quad x \geqslant x_{i+1} . \tag{13}$$

The behavior in the intervals $x_{i-1} \leqslant x \leqslant x_i$ and $x_i \leqslant x \leqslant x_{i+1}$ is determined by the requirement that

$$\frac{d^2 S_i(x)}{dx^2} = 0 \tag{14}$$

except at mesh points.

The Galerkin method is now applied to Eqs. (8) and (9). If Eq. (12) is substituted into Eq. (8) and an integration is done over the transverse coordinates, then

UNCLASSIFIED

$$\sum_{i,j} C_{i,j}^{n+1} \iint dxdy\, S_\ell(x) S_m(y) \left(1 - i\,\frac{\Delta Z}{2}\, H_x\right) S_i(x) S_j(y)$$

$$= \sum_{i,j} C_{i,j}^{n} \iint dxdy\, S_\ell(x) S_m(y) \left(1 + i\,\frac{\Delta Z}{2}\, H_y\right) S_i(x) S_j(y). \tag{15}$$

Define

$$[S_x]_{i,\ell} \equiv \int dx\, S_\ell(x) S_i(x) \tag{16}$$

and

$$[H_x]_{i,\ell} \equiv \int dx\, S_\ell(x) H_x S_i(x),$$

so that Eq. (15) becomes

$$\sum_{i,j} C_{i,j}^{n+1} \left\{ [S_x]_{i,\ell} [S_y]_{j,m} - i\,\frac{\Delta Z}{2}\, [H_x]_{i,\ell} [S_y]_{j,m} \right\}$$

$$= \sum_{i,j} C_{i,j}^{n} \left\{ [S_x]_{i,\ell} [S_y]_{j,m} + i\,\frac{\Delta Z}{2}\, [H_y]_{j,m} [S_x]_{i,\ell} \right\} \tag{17}$$

or, in matrix notation,

$$\left\{ S_x \times S_y - i\,\frac{\Delta Z}{2}\, H_x \times S_y \right\} \mathbf{C}^{n+1} = \left\{ S_x \times S_y + i\,\frac{\Delta Z}{2}\, S_x \times H_y \right\} \mathbf{C}^{n}$$

operate with $S_x^{-1} \times S_y^{-1}$ to get

$$\left\{ S_x^{-1} S_x \times I - i\,\frac{\Delta Z}{2}\, S_x^{-1} H_x \times I \right\} \mathbf{C}^{n+1} = \left\{ I \times S_y^{-1} S_y + i\,\frac{\Delta Z}{2}\, I \times S_y^{-1} H_y^{-1} \right\} \mathbf{C}^{n} \tag{18}$$

where I is the unit matrix. Since the S and H matrixes will be shown to be of the band-diagonal form with elements

$$S_{i,k} = \sum_{n=-N}^{+N} b_n \delta_{i,k+n}$$

(see Eqs. (22) and (23), below), it follows that $S_x^{-1} H_x S_x = H_x$. Thus, Eq. (18) becomes

$$\left\{ S_x \times I - i\,\frac{\Delta Z}{2}\, H_x \times I \right\} S_x^{-1} \mathbf{C}^{n+1} = \left\{ I \times S_y + i\,\frac{\Delta Z}{2}\, I \times H_y \right\} S_y^{-1} \mathbf{C}^{n}. \tag{19}$$

The q, p element is

$$\sum_{i,j} \left\{ [S_x]_{q,i}\delta_{p,j} - i\,\frac{\Delta Z}{2}\,[H_x]_{q,i}\delta_{p,j} \right\} \sum_{\ell} [S_x^{-1}]_{i,\ell}\, C^n_{\ell,j}$$

$$= \sum_{i,j} \left\{ \delta_{q,i}[S_y]_{p,j} + i\,\frac{\Delta Z}{2}\,\delta_{q,i}[H_y]_{p,j} \right\} \sum_{\ell} [S_y^{-1}]_{j,\ell}\, C^n_{i,\ell}\,. \tag{20}$$

The matrix elements are derived by solving Eq. (14) for the $S_i(x)$ using the boundary conditions in Eq. (13). For uniform mesh spacing Δx the result is

$$S_i(x) = \frac{x_i - x}{\Delta x} \qquad x_{i-1} \leqslant x \leqslant x_i$$

$$S_i(x) = \frac{x_{i+1} - x}{\Delta x} \qquad x_i \leqslant x \leqslant x_{i+1}\,. \tag{21}$$

The integrations in Eq. (16) can now be performed to give

$$[S_x]_{i,j} = \int_{-\infty}^{\infty} dx\, S_i(x)\, S_j(x) = \Delta x \left(\frac{2}{3}\,\delta_{i,j} + \frac{1}{6}\,\delta_{i,j+1} + \frac{1}{6}\,\delta_{i,j-1} \right) \tag{22}$$

and

$$[H_x]_{i,j} = \int_{-\infty}^{\infty} dx\, S_i(x)\, \frac{d^2}{dx^2}\, S_j(x) = -\int_{-\infty}^{\infty} dx\, \frac{dS_i}{dx}\,\frac{dS_j}{dx}$$

$$= -\Delta x \left(\frac{2\delta_{i,j} - \delta_{i,j+1} - \delta_{i,j-1}}{(\Delta x)^2} \right). \tag{23}$$

After Eqs. (22) and (23) are inserted in Eq. (20) (and the equivalent equation for the second half of the ADI scheme), the sums are performed to give

$$\left(\frac{1}{6} - i\eta_x\right)(\alpha_x)^{n+1}_{q-1,p} + \left(\frac{2}{3} + 2i\eta_x\right)(\alpha_x)^{n+1}_{q,p} + \left(\frac{1}{6} - i\eta_x\right)(\alpha_x)^{n+1}_{q+1,p}$$

$$= \left(\frac{1}{6} + i\eta_y\right)(\alpha_y)^{n}_{q,p-1} + \left(\frac{2}{3} - 2i\eta_y\right)(\alpha_y)^{n}_{q,p} + \left(\frac{1}{6} + i\eta_y\right)(\alpha_y)^{n}_{q,p+1} \tag{24}$$

and

$$\left(\frac{1}{6} - i\eta_y\right)(\alpha_y)^{n}_{q,p-1} + \left(\frac{2}{3} + 2i\eta_y\right)(\alpha_y)^{n+2}_{q,p} + \left(\frac{1}{6} - i\eta_y\right)(\alpha_y)^{n+2}_{q,p+1}$$

$$= \left(\frac{1}{6} + i\eta_x\right)(\alpha_x)^{n+1}_{q-1,p} + \left(\frac{2}{3} - 2i\eta_x\right)(\alpha_x)^{n+1}_{q,p} + \left(\frac{1}{6} + i\eta_x\right)(\alpha_x)^{n+1}_{q+1,p} \tag{25}$$

where

UNCLASSIFIED

$$(\alpha_x)^n_{q,p} = \sum_{\ell} [S_x^{-1}]_{q,\ell} C^n_{\ell,p} \tag{26}$$

and

$$(\alpha_y)^n_{q,p} = \sum_{\ell} [S_y^{-1}]_{q,\ell} C^n_{\ell,p} \tag{27}$$

and η_x and η_y are defined just prior to Eq. (10). Equations (24) and (25) should be compared with Eqs. (10) and (11) for the CDA transverse discretization. Note that the algorithm is now in terms of coefficients $(\alpha_{x,y})^n_{g,p}$ inverse to the coefficients $C^n_{g,p}$. Thus, two additional steps are required in the elementary sequence described in Sec. 3. After the fields $\Phi(\tilde{x}, \tilde{y}, z)$ are formed they are identified with the coefficients $C^n_{q,p}$ from Eq. (12), since the solution is sought and data are sampled only at mesh points. Then the α coefficients are formed via Eqs. (26) and (27). This proceeds rapidly, however, since the S matrices are tridiagonal and rapidly inverted. After the new α's are determined by the algorithm of Eq. (24) or (25) the C's are reconstructed by, for example,

$$C^n_{q,p} = \sum_{\ell} [S_x]_{q,\ell} (\alpha_x)^n_{\ell,p}$$

using Eq. (22). Again, the identification of the new fields $\Phi(\tilde{x}, \tilde{y}, z)$ is made with the new C coefficients at mesh points so that the fields ψ_4 can be reconstructed with the inverse phase change $-0.5 \int g dz'$, as described in Sec. 3.

5. BOUNDARY VALUES

The problem has a line of symmetry for $x = 0$, $y = (-\infty, +\infty)$, that is, for the wind along the y axis. The other boundaries are at $y = +\infty$, all x; $y = -\infty$, all x; and $x = +\infty$, all y. Hence, we have nine regions, each of which must be handled differently. These are listed in Table 1 and depicted in Fig. 1. These regions are identified in the Fortran listing of subroutine ADVANCE (see Appendix).

Table 1
Nature of Regions Handled by PROP-I

Region	Subscripts		Nature
	I	J	
I	2, NX-1	2, NY-1	No boundary, no symmetry
II	1	2, NY-1	x symmetry
III	NX	2, NY-1	x boundary ($+\infty$)
IV	2, NX-1	1	y boundary ($-\infty$)
V	2, NX-1	NY	y boundary ($+\infty$)
VI	1	1	x symmetry, y boundary ($-\infty$)
VII	1	NY	x symmetry, y boundary ($+\infty$)
VIII	NX	NY	x boundary ($+\infty$), y boundary ($+\infty$)
IX	NX	1	x boundary ($+\infty$), y boundary ($-\infty$)

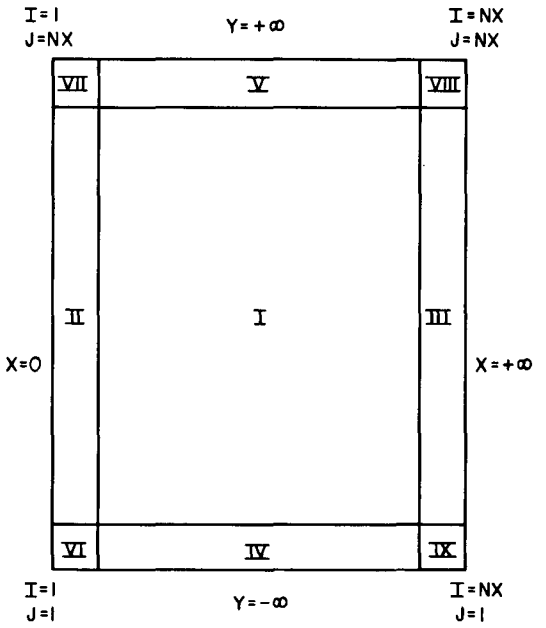

Fig. 1—Nine regions handled by program PROP-I

6. DESCRIPTION OF PROP-I

The main program begins with a brief description of the objective of the program itself. Further comment cards define the physical, mathematical, or numerical significance of the Fortran names ascribed to the variables. Other definitions and units are listed in Subroutine INPUT (see below). Since the comment cards in the listing provide very terse definitions of the quantities involved, these are amplified in some measure here.

6.1. Common Arrays

In the computation, at each Z-plane, the values of various quantities at each point in the grid spanning the area transverse to the beam are stored in common arrays. These arrays are

CC: This array indicates changes in the index of refraction, that is, the value of the entire coefficient of ψ_4 in the nonlinear term of Eq. (5). (See Subroutine INDEX in the appended Fortran listings.)

PLTGSN: This array calculates the parameter N of Gebhardt and Smith (7).

PLTRAT: This array stores the ratio of the average intensity inside the isophote of the bloomed beam to that of the vacuum beam multiplied by the absorption factor $a^{-\alpha Z}$. Isophote μ is that contour whose irradiance is the peak irradiance of the beam multiplied by μ; μ here is taken to be 0.2, 0.5, and 0.8. These values are plotted at the conclusion of the calculation (see Graph 2) but printed out in tabular form at select ranges. (See Subroutine OUTPUT and Subroutine Graph 2 in the Fortran listings.)

U, V: $\psi(\tilde{x}, \tilde{y}, \tilde{Z}) \equiv u(\tilde{x}, \tilde{y}, \tilde{Z}) + iv(\tilde{x}, \tilde{y}, \tilde{Z})$.

UNCLASSIFIED

6.2. Common Variables and Constants

D: $D(z) = \dfrac{z^2}{k^2 a_0^4} + \left(1 - \dfrac{z}{f}\right)^2 .$

FOCUS: $\sqrt{D}$.

HZ: The program equivalent of $\Delta\tilde{Z}$.

HX: The program equivalent of $\Delta\tilde{X}$.

HY: The program equivalent of $\Delta\tilde{y}$.

IMAX and JMAX: Indexes (I, J) label the points in the arrays transverse to the beam axis. $I = 1, 2, \ldots,$ NX + 1, $J = 1, 2, \ldots,$ NY + 1; (IMAX, JMAX) locates the point in the grid at which the beam intensity reaches its peak.

NOUT: If NOUT = 1, the program will abort when $|\iint d\tilde{x}d\tilde{y}|\psi|^2 - 1| <$ ECHNG; see INPUT for ECHNG. If NOUT $\neq$ 1, this option is not exercised, and the program will continue to compute even though this relation ceases to be satisfied.

SUM: $Pe^{-\alpha z}\iint d\tilde{x}d\tilde{y}|\psi|^2$ = total integrated intensity at z.

WIDTH: The quantity a_0.

W2: The quantity a_0^2.

ZETA: The quantity z/ka_0^2.

Items listed in the main program but not discussed here are regarded as self-evident.

6.3. Subroutines

The main program functions as an executive routine, the detailed computations being relegated to subroutines. Iteration in the Z variable is also done within the main program as well as tests on whether or not energy is conserved to within prescribed tolerances (see NOUT, above).

Subroutine INPUT: This subroutine reads the input data cards; this listing contains comment cards that explain the entries. Some added comments are appropriate here.

HX and HY correspond to $\Delta\tilde{x}$ and $\Delta\tilde{y}$. For Gaussian beams, HX, HY $\sim$ 0.18 have been found to be adequate for distortions that are not too severe; in the latter case, more mesh points are needed (that is, larger NX, NY and, often, concomitantly smaller values of HX and HY).

KQMAX determines the number of times detailed information on I_{rel}, $\langle I_{\text{rel}}\rangle_\mu$, etc., are printed.

LPLOT: Group plotting means that the isoirradiance contour plots for *six* ranges will be plotted on one sheet of plotting paper, to the same scale. Where many runs are being done, this is a space-saving device. Sequential plotting means that the contour plots (and others) are each plotted separately.

NFOCUS: When sequential plotting is used, plotting may be done at all ranges on the same scale as at the aperture (nonfocused plotting) or magnified (focused plotting).

ZT: For $Z > ZT$, the absorption coefficient is set equal to zero. This is useful for comparisons with experiment and for theoretical study.

DIAM: DIAM = $2\sqrt{2a_0}$ is regarded as the diameter of the optics (even for an infinite Gaussian beam). This is a definition, otherwise, of a_0 as it appears in the theory.

THETA: For beams in atmosphere not parallel to the surface; read in degrees. (Appropriate calculations of atmospheric variables as a function of altitude must be added to Subroutine UPDATZ (see below) to use this angle.)

WN = $k = 2\pi/\lambda$.

Since the determination of the absorption coefficient through the specification of P, PH20, and T by the code (see AHV and Subroutine SETUP) is not always convenient or appropriate, a set of options are allowed. These are provided for on Card 5 when the quantities δ, τ connected with kinetic cooling in CO_2 laser beams, total absorption coefficient α, and dimensionless constant β may be separately specified, replacing those calculated by the code from data on Cards 1-4.

INPUT: Prints the input data.

Subroutine SETUP: See listing.

Subroutine INITL: This subroutine calculates the real and imaginary part of the beam amplitude at the aperture ($\tilde{Z} = 0$) for each point in the grid and stores the values in the appropriate array. This subroutine needs the initial beam data provided by Subroutine VACAMP.

Subroutine VACAMP: This subroutine computes the vacuum beam everywhere; in particular it is called upon by Subroutine INITL to set up the boundary conditions for the problem. In addition, while Subroutine INTENS does not explicitly call upon Subroutine VACAMP for calculating the quantity VACINT, which is the intensity of the vacuum beam corrected for absorption, it should be noted that in the present listing, Subroutine VACAMP and the computations for the table CONMIN assume the vacuum beam to be Gaussian. For a non-Gaussian beam, the formula for U and V must be changed accordingly, or the entries be provided for in a table. In the latter case, the CONMIN table should be removed, since it becomes meaningless, or corrected, if the vacuum beam at non-zero values of Z is determined analytically.

UNCLASSIFIED

Subroutine ADVANCE: This is the ADI method, using CDA transverse discretization, if NDIF = 0, otherwise the Galerkin method with linear splines.

Subroutine UPDATZ: See listing for comment cards.

Subroutine BOUNDS: Assign boundary values to U and V at the edges of the grid; these are taken to be zero, except *downwind*. When the beam is sufficiently weak at the boundaries, the calculation is insensitive to assigning boundary values of zero.

Subroutine SHIFT: See listing.

Subroutine FOLLOW: This subroutine displaces the values in the arrays for U and V and intensity so as to keep the peak intensity point in the beam in the center of the grid.

CAUTION: This subroutine is incorrect when an array has an explicit dependence on the variable Y; this case occurs when kinetic cooling is present, that is, $\delta \neq 0$, $\tau \neq 0$. However, it has been determined experimentally that τ is substantially smaller than had been thought (8). Therefore, for CO_2 laser beams, δ has been consistently put equal to zero; for beams other than CO_2, the kinetic cooling theory is inappropriate anyway, so again δ is set equal to zero. The subroutine, under these conditions, correctly shifts the arrays.

Subroutine INTENS: See comment cards in listing. Subroutine ITENS also prints the array that compares intensities averaged over isoirradiance contours with the corresponding vacuum beam quantities, the areas of these regions, etc. (See comments, above, in Subroutine VACAMP.)

Subroutines GRAPH 1, GRAPH 2, TOPOGRAF, INFORMAT: See listing for comments.

This program has been written to be used on the CDC 3800 at the Naval Research Laboratory. The last four subroutines call upon software that will vary with the installation.

7. CONCLUSIONS

This report has presented in detail the development of an algorithm to advance rapidly the solution of a partial differential equation which describes intense laser light propagation in an absorbing medium. Two separate discretization schemes were discussed: Central differencing and a Galerkin method with linear spline basis functions. The two methods give essentially equivalent results for a wide range of cases of interest. The Galerkin method has, in addition, the inherent ability to handle strong discontinuities in the transverse distribution of the irradiance, such as would be produced by an aperture. The effect of the Galerkin method with splines is to smooth out high frequency contributions which tend to be amplified in the central difference approximation.

ACKNOWLEDGMENT

The author gratefully acknowledges helpful discussions with J.N. Hayes and E.H. Takken.

REFERENCES

1. A.H. Aitken, J.N. Hayes, and P.B. Ulrich, "Propagation of High-energy 10.6-micron Laser Beams Through the Atmosphere," NRL Report 7293, May 28, 1971. AD 72511.

2. P.B. Ulrich, "A Numerical Calculation of Thermal Blooming of Pulsed, Focused Laser Beams," NRL Report 7382, Dec. 30, 1971.

3. J. Herrmann and L.C. Bradley, "Numerical Calculation of Light Propagation," Massachusetts Institute of Technology, Lincoln Laboratories, Lexington, Massachusetts, Report LTP-10, July 12, 1971.

4. D.W. Peaceman and H.H. Rachford, "The Numerical Solution of Parabolic and Elliptic Differential Equations," *J. SIAM* 3, 28 (1955).

5. L.V. Kantorovich and V.I. Krylov, *Approximate Methods of Higher Analysis, Interscience*, New York, 1958, p. 261.

6. G.E. Forsythe and W.R. Wasow, *Finite Difference Methods for Partial Differential Equations*, Wiley, New York, 1960, p. 104.

7. F.G. Gebhardt and D.C. Smith, "Self-induced Thermal Distortion in the Near Field for a Laser Beam in a Moving Medium," *IEEE J. Quantum Electron.* QE-7, 63 (1971).

8. L. Sica, *Appl. Phys. Latt.* 22, 396 (1973).

UNCLASSIFIED

Appendix A
FLOWCHART AND FORTRAN LISTINGS

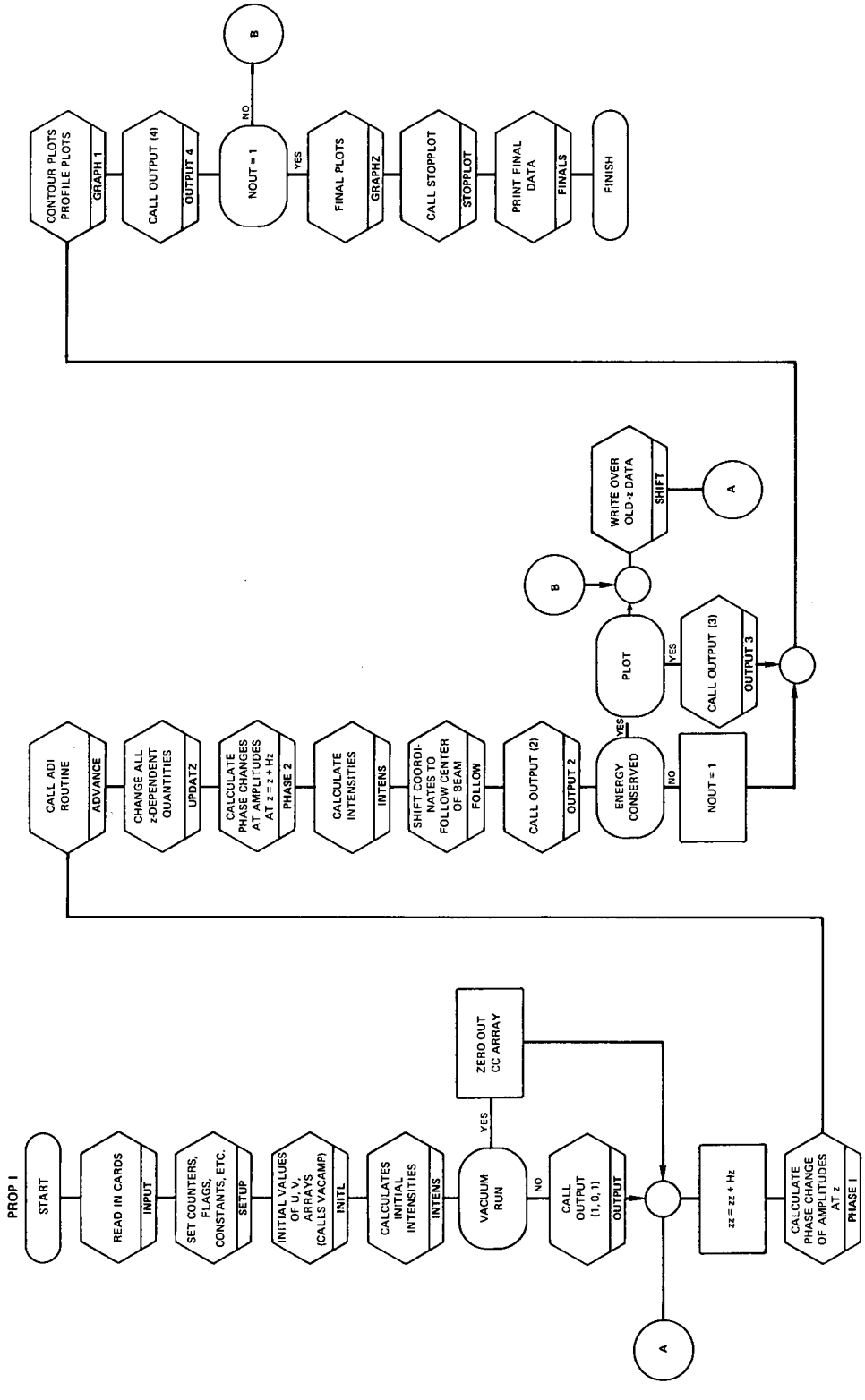

Fig. A1—Flowchart for program PROP-I

UNCLASSIFIED

```
      PROGRAM PROP I

C*********************************************************************

C       PROGRAM PROP IS A THREE DIMENSIONAL, TIME INDEPENDENT
C       CODE DESCRIBING THE STEADY STATE PROPAGATION OF HIGH
C       INTENSITY, UNPOLARIZED MONOCHROMATIC LIGHT IN A GASEOUS
C       MEDIUM IN WHICH A WIND IS PRESENT, THE BEAM CAN BE INITIALLY
C       FOCUSSED OR CAN BE LAUNCHED PARALLEL TO THE Z AXIS
C       THE INTENSITY OF THE LIGHT BEAM IS GREAT ENOUGH TO AL-
C       TER THE DENSITY OF THE GAS THROUGH DIRECT HEATING AND THROUGH
C       COOLING BY MEANS OF THE VIBRATIONAL-TRANSLATIONAL TIME LAG, THE
C       ALTERED DENSITY IMPLIES A CHANGE OF INDEX OF REFRACTION WHICH
C       IN TURN ALTERS THE LIGHT PATH SINCE THE INDEX APPEARS AS A FACTOR
C       IN THE WAVE EQUATION, FURTHERMORE, SINCE THE INTENSITY DEPENDS ON
C       THE SQUARE OF THE LIGHT AMPLITUDE, THE PROBLEM IS NON-LINEAR,
C       THE APPROPRIATE EQUATION IS THE SCALAR  HELMHOLTZ  EQUATION TOGE-
C       THER WITH THE PARAXIAL (FRESNEL) APPROXIMATION WHICH STATES
C       THAT THE SECOND DERIVATIVE OF THE AMPLITUDE WITH RESPECT TO Z
C       IS MUCH LESS THAN WAVE NUMBER TIMES FIRST DERIVATIVE WITH
C       RESPECT TO Z,  THE RESULTING EQUATION IS PARABOLIC AND IS THERE-
C       FORE UNIQUELY SOLVED BY INITIAL DATA AT THE SOURCE PLANE,  HENCE
C       A MARCHING TECHNIQUE IS USED TO PROCEED FROM Z=0 TO DISTANCES
C       OF INTEREST (1 TO 10 KILOMETERS)


C     COMMON ARRAYS

C        CC          INDEX OF REFRACTION CHANGES
C        FORM1       PRINTING STATEMENT FORMATS
C        FORM2       PRINTING STATEMENT FORMATS
C        PLTRAT      RATIOS OF INTENSITY TO VACUUM INTENSITY FOR PLOTTING
C        QI          INTENSITY IN A TRANSVERSE PLANE AS A FUNCTION OF
C                    XI AND ETA
C        QJMAX       BEAM DEFLECTIONS
C        QQMAX       MAXIMUM INTENSITY VALUES
C        TENS        INTEGRATED INTENSITY - CALCULATED AT EACH STEP
C        U,V         REAL,IMAGINARY PARTS OF THE AMPLITUDE
C        XT          Z VALUES FOR PLOTTING OF TENS (IN KILOMETERS)
C        ZI          POINTS AT WHICH INTENSITY RATIOS ARE CALCULATED
C        ZM          POINTS AT WHICH INTENSITY MAXIMA AND DEFLECTIONS
C                    ARE CALCULATED ,,,,,,,,IN KILOMETERS

C     COMMON VARIABLES AND CONSTANTS GENERATED BY THE PROGRAM

C     AAA    = INTEGRATION FACTOR, =HZ/(HX*HX)/16
C     ALFCON = STORES INITIAL VALUE OF ALPHA
C     ALFSUM = LOCATION FOR ALPHAINTEGRATION SUMMING DURING Z-ITERATIONS
C     BBB    = INTEGRATION FACTOR, =HZ/(HY*HY)/16
C     BETAZ  = SAVES INITIAL VALUE OF BETA
C     CZERO  = STORES INITIAL VALUE OF C
C     DDD    = INTEGRATION FACTOR, =HZ/HX/16
C     EEE    = INTEGRATION FACTOR, =HZ/HY/16
C     FOCUS  = FACTOR FOR SHRINKING COORDINATE SYSTEM
```

```
C       HZV     = STORES INITIAL HZ VALUE
C       IMAX AND JMAX = INDICES OF LARGEST QI(I,J), (PEAK INTENSITY)
C       IQ      = INDEX FOR INTENSITY MAXIMUM (QQMAX)
C       IT      = INDEX FOR TOTAL INTEGRATED INTENSITY (TENS)
C       JCMAX   = J-INDEX OF LARGEST CI(NX,J) FOR GIVEN NX
C       JMAX      (SEE IMAX)
C       KJ      = COUNTER TO BE TESTED FOR SKIPPING PLOTS
C       KKMAX   = NUMBER OF STEPS BETWEEN MAXIMUM INTENSITY CALCULATIONS
C       KP      = COUNTER FOR INTENSITY MAXIMUM CALCULATION
C       NAPLOT  = NUMBER OF STEPS BETWEEN PLOTS
C       NCUT    = ENERGY ABORT FLAG
C       NX1     = NUMBER OF CELLS IN X-DIRECTION
C       NX2     = CENTRAL MESH POINT IN X-DIRECTION
C       NXY     = COUNTS CALL TO TOPOGRAF
C       NY1     = NUMBER OF CELLS IN Y-DIRECTION
C       NY2     = CENTRAL MESH POINT IN Y-DIRECTION
C       PI      = 3.1415926
C       PLTCAL  = GIVES PARAMETER PRINTOUT ON 1ST TOPOGRAF PLOT
C       ROZERO  = STORES INITIAL VALUE OF RO
C       SUM     = TOTAL INTEGRATED INTENSITY
C       SUM1    = SAVES INITIAL SUM FOR SUM2 CALULATION
C       TEMP    = AMBIENT TEMPERATURE IN DEGREES CENTIGRADE
C       WIDTH   = INITIAL RADIUS
C       W2      = INITIAL RADIUS SQUARED
C       Z       = DISTANCE ALONG BEAM PATH IN KILOMETERS
C       ZZ      = DISTANCE ALONG BEAM PATH  (IN PROGRAM UNITS)


      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NAPLOT,NOLT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH2O,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON/NNN/NFOCUS

      BANK,(0), /1/

      CALL INPUT
      CALL SETUP
      CALL INITL (1)
      CALL INTENS
C     TEST FOR VACUUM RUN
      IF (BETA.EQ.0) GO TO 1
      GO TO 4
    1 PLTCAL=2
      PRINT 2
    2 FORMAT (21H1THIS IS A VACUUM RUN )
      ALPHA=0.0
      DO 3 I=1,NX
      DO 3 J=1,NY
    3 CC(I,J) = 0
```

UNCLASSIFIED

```
    4 CALL OUTPUT (1,0,1)

      KP=KP-1
C     BEGIN ITERATIONS IN Z

      DEFL=0.
      QGM=0.
      DO 8 JJ=1,NSTEP
      NMOD=JJ-((JJ-1)/2)*2

C     COUNT NUMBER OF ITERATIONS BETWEEN PLOTS WHILE IN PLOTTING AREA
      IF (STARTT.LE.Z .AND.Z .LE.STOPP) KJ=KJ+1
      KP=KP+1
C     FOR FINAL Z FORCE PLOT AND MAXIMUM INTENSITY CALCULATION
      IF(JJ.NE.NSTEP) GO TO 41
      KP=KKMAX
      KJ=NNPLOT
41    CONTINUE
      CALL PHASE(1,JJ)
      CALL ADVANCE(NMOD)
      CALL UPDATZ
      CALL PHASE(2,JJ)
      CALL INTENS
      CALL FOLLOW(DEFL)
      PRINT 100,QI(1,NY2),Z
100   FORMAT(* ON-AXIS INTENSITY =*,F15.11,*       Z=*,F10.3,*KM*)

      CALL OUTPUT (2,JJ,0)
      IF(ABSF((SUM-POWER)/POWER).LT.ECHNG) GO TO 11
      NOUT=1
      NXY=6
      GO TO 6
11    CONTINUE

C     TEST COUNTER FOR PLOTTING
      IF (KJ.NE.NNPLOT) GO TO 7
      CALL OUTPUT(3,JJ,2)

6      CALL GRAPH1(NFOCUS)
      CALL OUTPUT (4,JJ,0)
      IF(NOUT.EQ.1) GO TO 9

7     CALL SHIFT
    8 CONTINUE

C     Z ITERATIONS ARE FINISHED,  DO FINAL GRAPHS

9     CONTINUE
      CALL GRAPH2
      CALL STOPPLOT

C     PRINT INPUT DATA AND SELECTED OUTPUT (IN DUPLICATE)
      CALL FINALS

      STOP
      END
```

```
      SUBROUTINE INPUT

C     DATA CARDS READ SHOULD CONTAIN THE FOLLOWING

C     CARD 1

C        HX,HY,HZ  STEP SIZES IN THE X,Y,Z DIRECTIONS (DIMENSIONLESS)
C        DIAM      FULL WIDTH OF INITIAL GAUSSIAN PROFILE
C                  OR DIAMETER (E**-2 POWER) OF APERTURE (CM)
C        WN        WAVE-NUMBER OF LIGHT SOURCE (CM-1)
C        F         FOCAL LENGTH OF FOCUSSED BEAM (CM)
C        ZT        DISTANCE (KM) AT WHICH INTERACTION STOPS,

C     CARD 2

C        NX        NUMBER OF MESH POINTS IN X DIRECTION
C        NY        NUMBER OF MESH POINTS IN Y DIRECTION
C        NSTEP     NUMBER OF ITERATIONS TO END OF RUN
C        NPLOT     NUMBER OF PLOTS
C        KOMAX     NUMBER OF MAX INTENSITY CALCULATIONS
C        STARTT    DISTANCE(IN KILOMETERS) AT WHICH PLOTTING STARTS
C        STOPP     DISTANCE(IN KILOMETERS) AT WHICH PLOTTING STOPS
C        LPLOT     = 0  FOR GROUP PLOTTING
C                  = 1  FOR SEQUENTIAL PLOTTING
C        NFOCUS    =0 FOR FOCUSED PLOTTING
C                  =1 FOR NON-FOCUSSED PLOTTING
C        NDIF      = 0  FOR CENTRAL DIFFERENCE TRANSVERSE DISCRETIZATION
C                  = 1  FOR GALERKIN METHOD WITH LINEAR SPLINE BASIS FOR
C                  TRANSVERSE DISCRETIZATION

C     CARD 3

C        GAMMA     RATIO OF SPECIFIC HEATS
C        REFRAC    MOLECULAR REFRACTIVITY = .154 CM3/GM,FOR AIR, AT 10.6
C                  MICRONS
C        C         VELOCITY OF SOUND   CM/SEC
C        VZERO     TRANSVERSE WIND VELOCITY (CM/SEC)
C        POWER     INPUT POWER IN WATTS
C        THETA     ELEVATION ANGLE OF BEAM (DEGREES)

C     CARD 4

C        PH2O      PARTIAL PRESSURE OF WATER IN TORR
C        P         TOTAL AMBIENT PRESSURE IN TORR
C        OMEGA     SLUING RATE IN RADIANS/SEC
C        ECHNG     ABSOLUTE MAGNITUDE OF RELATIVE CHANGE OF INTEGRATED
C                  INTENSITY TO CAUSE RUN ABORT
C        RO        AMBIENT DENSITY, RHO (GM/CM3)
C        TEMP      AMBIENT TEMPERATURE IN DEGREES CENTIGRADE

C     CARD 5

C        NDEL      SET =1 TO ENTER VALUE OF DELTA THAT FOLLOWS,
C        DELTA     COOLING PARAMETER (DIMENSIONLESS)
C        NTAU      SET =1 TO ENTER VALUE OF TAU THAT FOLLOWS,
```

UNCLASSIFIED

```
C         TAU        NITROGEN V-T RELAXATION TIME (SEC)
C         NALPH      SET =1 TO ENTER VALUE OF ALPHA THAT FOLLOWS.
C         ALPHA      TOTAL ABSORPTION COEFFICIENT (CM-1)
C         NBETA      SET =1 TO ENTER VALUE OF BETA THAT FOLLOWS.
C         BETA       FACTOR BEFORE INTEGRAL OF INTENSITY OVER WIND
C                    DIRECTION IN SUBROUTINE INDEX (DIMENSIONLESS)

C      CARD 6

C         CASE       LASER TYPE


      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOLT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON/NNN/NFOCUS
      COMMON /TYPE/ CASE(4)

      BANK,(0), /1/

      READ 100,HX,HY,HZ,DIAM ,WN,F,ZT
100   FORMAT(8E10.3)
      JSEQ=MPRISEQ(1)
      CALL DATE(MONTH,IDAY,IYEAR,JULDAY)
      PRINT 101,JSEQ,MONTH,IDAY,IYEAR
  101 FORMAT(* SEQUENCE *04,8X,I2*/*I2*/*I2/)

      READ 200,NX,NY,NSTEP,NPLOT,KQMAX,STARTT,STOPP,LPLOT,NFOCUS,NDIF
200   FORMAT(5I5,2E10.3,3I5)
      PRINT 201,HX,HY,HZ,DIAM ,WN,F,  NX,NY,NSTEP,NPLOT,KQMAX,LPLOT,NDIF
  201 FORMAT(4H HX=E12.5/4H HY=E12.5/4H HZ=E12.5/6H DIAM=E12.5/
     14H WN=E12.5/3H F=E12.5/            4H NX=I5/4H NY=I5/
     17H NSTEP=I5/7H NPLOT=I5/7H KQMAX=I5/7H LPLOT=I5/6H NDIF=I5//)
      PRINT 202,STARTT,STOPP
  202 FORMAT(19H PLOTTING STARTS AT,F10.5,24H KILOMETERS AND STOPS AT,
     1F10.5,11H KILOMETERS/)

      READ 300,GAMMA,REFRAC,C,VZERO,POWER,THETA
300   FORMAT(6E10.3)
      PRINT 301,GAMMA,REFRAC,C,VZERO,POWER,THETA
      THETA=THETA*(3.14159654/180.)
  301 FORMAT(7H GAMMA=E12.5/8H REFRAC=E12.5/3H C=E12.5/
     17H VZERO=E12.5/7H POWER=E12.5//7H THETA=E12.5//)

      READ 400,PH20,P,OMEGA,ECHNG,RO,TEMP
  400 FORMAT(6E10.3)
      PRINT 401,PH20,P,OMEGA,ECHNG,RO,TEMP
  401 FORMAT(6H PH20=E12.5/3H P=E12.5/7H OMEGA=E12.5///,7H ECHNG=E12.5
     1//* ROZERO=*E12.5//* TEMP=*E12.5//)
```

```
      READ 500,NDEL,DELTA,NTAU,TAU,NALPH,ALPHA,NBETA,BETA
  500 FORMAT(4(I5,E10.3))

      READ 600,(CASE(I),I=1,4)
  600 FORMAT(4A8)
      PRINT 601, CASE
  601 FORMAT(14H LASER TYPE:   ,4A8)

      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE SETUP

C     SET FLAGS, COUNTERS, INDICES, CONSTANTS, AND STEP SIZES

      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SLM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON/ABC/A(61),B(61),E(61)
      DATA (A=61(.166666)),(B=61(.6666666)),(E=61(.1666666))

      TYPE INTEGER FORM1, FORM2

C     PARRAY PROVIDES WORKING LOCATIONS FOR SYSTEM PLOTTING ROUTINES
      DIMENSION PARRAY(1000)

      BANK,(0), /1/

C      INITIALIZE SYSTEM PLOTTER
      CALL PLOTS(PARRAY,1000,1)

      PI=3.1415926
      Z=0.
      ZETA=0.
      ZZ=0.
      D=1.
      FOCUS=1.
      IT=1
      IG=1
      NXY=1
      NOUT=0
      HZV=HZ
      CZERO=C
      ROZERO=RO
      KP=KKMAX
      NX1=NX-1
      NY1=NY-1
      NX2=NX/2+1
      NY2=NY/2+1
      KKMAX=(NSTEP-KQMAX)/KQMAX+1
      WIDTH=DIAM/(2.0*SQRT(2.0))
      W2=WIDTH*WIDTH
      FL=F/100000.
      HH=HZ*W2*WN/100000.
      PLTCAL=1.

      NNPLOT=NSTEP/(NPLOT-1)-1
      IF(STARTT.GT.0.0) NNPLOT=((STOPP-STARTT)/HH)/(NPLOT-1)-1
      KJ=NNPLOT-1
```

```
      CALL PLOT(,8,0,*3)

C     CALCULATE COOLING PARAMETERS IF NOT READ IN
C     ALPHA2 = ABSORPTION COEFFICIENT OF CO2

      ALPHA2=
     1  0,00144*((295,/(273,0+TEMP))**1,5)*((10,)**(-970,0/(TEMP+273,)))
      IF (NTAU,EQ,1) GO TO 1
      TAU=1,/(30,0*38,0*PH2O)

    1 IF (NALPH,EQ,1) GO TO 2
      ALPHA=(4,32E-011)*PH2O*(P+193,*PH2O) + ALPHA2

    2 IF (NDEL,EQ,1) GO TO 3
      DELTA=(ALPHA2/ALPHA)*2,44

    3 PRINT 4, TAU,ALPHA,ALPHA2,DELTA
    4 FORMAT(5H TAU=E12,5/7H ALPHA=E12,5/8H ALPHA2=E12,5/7H DELTA=E12,5)


C     CALCULATE BETA

      IF (NBETA,EQ,1) GO TO 11
      BETA =(GAMMA-1,)*ALPHA*REFRAC*3,*POWER*(1,0E+007)*WIDTH*WN*WN
     1/(C*C*VZERO)
   11 PRINT 12, BETA
   12 FORMAT (6H BETA= E12,5///)
      BETAZ=BETA

C      CALCULATE POSITION AND SIZE OF VACUUM BEAM WAIST,
C      AND SIZE OF BEAM AT FOCAL POINT

      ZWAIST=F/(1,+F*F/(WN*WN*W2*W2))
      DWAIST=(ZWAIST/(WN*W2))**2,*(1,-ZWAIST/F)*(1,-ZWAIST/F)
      DF=(F/(WN*W2))**2,
      PRINT 50,ZWAIST,DWAIST,DF
50    FORMAT(* ZWAIST=*E12,5/* D(WAIST)=*E12,5/* D(F)=*E12,5/)

C     SUPPLY TWO PRINTING FORMATS

      FORM1(1)= 8H(40(1X,I
      FORM1(2)= 3H2))
      FORM2(1)= 8H(17(1X,F
      FORM2(2)= 5H6,3))

   14 RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE UPDATZ

C     UPDATE Z FOR EACH ITERATION


      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPF,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ

      BANK,(0), /1/

      ZZ=ZZ+HZ
      FHAT=F/(WN*W2)
      ZETA=(TANF(ZZ-ATANF(WN*W2/F))+WN*W2/F)/(1.+WN*WN*W2*W2/(F*F))
      Z=ZETA*WN*W2/100000.
      Z1=1.-100000.*Z/F
      D=ZETA*ZETA+Z1*Z1
      FOCUS=SQRTF(D)

C      CALCULATE PRESSURES AND TEMPERATURES FOR SLANT BEAM PATH

      IF (THETA.EQ.0.0) GO TO 2
      HEIGHT=Z *SINF(THETA)
      PP=P*EXPF(-0.13*HEIGHT)
      PPH20=PH20*EXPF(-.54*HEIGHT)
      T=290.0-6.5*HEIGHT
      ALPHA2=1.44E-3*((295.0/T)**1.5)*((10.0)**(-970.0/T))
      ALPHA=4.32E-11*PPH20*(PP+193.0*PPH20)*ALPHA2
      RO = ROZERO-0.11E-3*HEIGHT
      C=CZERO-400.0*HEIGHT
      DELTA=(ALPHA2/ALPHA)*2.44
      TAU=1./((11.45+1.11E+4*PPH20/PP)*(PP/T))
    2 CONTINUE

C     TEST FOR END OF INTERACTION
      IF(Z.LT.ZT) GO TO 4
      BETA = 0
      ALPHA=0.
      DO 3 I=1,NX
      DO 3 J=1,NY
    3 CC(I,J)=0.0

    4 RETURN
      END
```

```
      SUBROUTINE BOUNDS(KK)

C      ASSIGN BOUNDARY VALUES

      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,      KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH2O,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ

      BANK,(0), /1/

      Y=(NY2-1)*HY
      DO 1 I=1,NX
      U(I,NY,KK)=0.
      V(I,NY,KK)=0.
      QI(I,NY)=0.
      CC(I,NY)=0.
1     CONTINUE
      DO 2 I=1,NX
      V(I,1,KK)=0.
      U(I,1,KK)=0.
      CC(I,1)=0.
      QI(I,1)=0.
2     CONTINUE
      DO 3 J=1,NY
      U(NX,J,KK)=0.
      V(NX,J,KK)=0.
      CC(NX,J)=0.
      QI(NX,J)=0.
3     CONTINUE

      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE SHIFT
      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ

      BANK,(0), /1/

C      SHIFT ALL STORED AMPLITUDES BACK ONE LOCATION IN Z BEFORE
C      PROCEEDING TO NEW Z-POINT


      DO 1 I=1,NX
      DO 1 J=1,NY
      U(I,J,1)=U(I,J,2)
      V(I,J,1)=V(I,J,2)
1     CONTINUE

      RETURN
      END
```

```
      FUNCTION VACAMP(X,Y,N)
      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOLT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH2O,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SLM1,TAL,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ

      BANK,(0), /1/

      ARG   =EXPF(-.5*(X*X+Y*Y))/SQRTF(PI)
      GO TO (1,2) N
1     VACAMP=ARG
      GO TO 100
2     VACAMP=0.
100   CONTINUE
      END
```

UNCLASSIFIED

```
      FUNCTION ERF(A,N)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNO,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOLT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPF,SUM,SLM1,TAL,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      GO TO (10,20)  N
10    ERF=1.0
      GO TO 100
20    ERF=HZ/(HX*HX)
100   RETURN
      END
```

```
      SUBROUTINE ADVANCE(N)
      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,T,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPF,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON/ABC/A(61),B(61),E(61)
      COMMON/XY/DEF
      DIMENSION CSR(61),CSI(61),DSR(61),DSI(61)
      DIMENSION DU(61),DV(61)
      TYPE REAL IMMY1,IMMNY,IMMNX

      BANK,(0), /1/

      NX11=NX1-1
      NY11=NY1-1
      Y=-HY*(NY2-1)+DEF
      REMY1=ERF(Y,1)
      Y=HY*(NY2-1)+DEF
      REMNY=ERF(Y,1)
      X=HX*(NX-1)
      REMNX=ERF(X,1)

      GO TO (100,200) N
100   CONTINUE

C**************CONSTANTS************************

      IMMNX=ERF(X,2)
      IMMNY=-ERF(Y,2)
      IMMY1=-ERF(Y,2)
      CRY=CRX=ARY=ARX=1./6.
      IF(NDIF.EQ.0) CRY=CRX=ARY=ARX=0.0
      BRY=BRX=2./3.
      IF(NDIF.EQ.0) BRY=BRX=1.0
      CIY=AIY=HZ/(2.*HY*HY)
      CIX=AIX=-HZ/(2.*HX*HX)
      BIY=-HZ/(HY*HY)
      BIX=HZ/(HX*HX)
C*********************************************

      IF(NDIF.EQ.0) GO TO 2003
      E(1)=1./6.
      DO 2001 I=1,NX
      DO 2000 J=1,NY
      DU(J)=U(I,J,1)
2000  DV(J)=V(I,J,1)
      CALL MXTRID(NY,DU)
      CALL MXTRID(NY,DV)
      DO 2002 J=1,NY
      U(I,J,1)=DU(J)
```

UNCLASSIFIED

```
2002  V(I,J,1)=DV(J)
2001  CONTINUE
2003  CONTINUE
C*************

C              THIS IS REGION ***VI***  J=1,I=1

      D1R=REMY1*U(1,1,1)-IMMY1*V(1,1,1)+CRY*U(1,2,1)-CIY*V(1,2,1)
      D1I=IMMY1*U(1,1,1)+REMY1*V(1,1,1)+CIY*U(1,2,1)+CRY*V(1,2,1)
      BRIX=1./(BRX*BRX+BIX*BIX)
      CSI(1)=2.*(CIX*BRX-CRX*BIX)*BRIX
      CSR(1)=2.*(CRX*BRX+CIX*BIX)*BRIX
      DSR(1)=(D1R*BRX+D1I*BIX)*BRIX
      DSI(1)=(D1I*BRX-D1R*BIX)*BRIX

C*************

C              THIS IS REGION ***IV***  J=1,I=2,NX1

      DO 2 I=1,NX11
      P=BRX-ARX*CSR(I)+AIX*CSI(I)
      Q=BIX-AIX*CSR(I)-ARX*CSI(I)
      PQ=1./(P*P+Q*Q)
      DIP1R=REMY1*U(I+1,1,1)-IMMY1*V(I+1,1,1)+CRY*U(I+1,2,1)
     1-CIY*V(I+1,2,1)
      DIP1I=IMMY1*U(I+1,1,1)+REMY1*V(I+1,1,1)+CIY*U(I+1,2,1)
     1+CRY*V(I+1,2,1)
      R=DIP1R+AIX*DSI(I)-ARX*DSR(I)
      S=DIP1I-AIX*DSR(I)-ARX*DSI(I)
      CSR(I+1)=(CRX*P+CIX*Q)*PQ
      CSI(I+1)=(CIX*P-CRX*Q)*PQ
      DSR(I+1)=(R*P+S*Q)*PQ
      DSI(I+1)=(S*P-R*Q)*PQ
2     CONTINUE

C*************

C              THIS IS REGION ***IX***  J=1,I=NX
      DNXR=REMY1*U(NX,1,1)-IMMY1*V(NX,1,1)+CRY*U(NX,2,1)-CIY*V(NX,2,1)
      DNXI=IMMY1*U(NX,1,1)+REMY1*V(NX,1,1)+CIY*U(NX,2,1)+CRY*V(NX,2,1)
      P=REMNX+AIX*CSI(NX1)-ARX*CSR(NX1)
      Q=IMMNX-AIX*CSR(NX1)-ARX*CSI(NX1)
      PQ=1./(P*P+Q*Q)
      R=DNXR+AIX*DSI(NX1)-ARX*DSR(NX1)
      S=DNXI-AIX*DSR(NX1)-ARX*DSI(NX1)
         DU(NX)=(R*P+S*Q)*PQ
         DV(NX)=(S*P-R*Q)*PQ
      DO 3 I=1,NX1
      NN=NX-I
      DU(NN)=DSR(NN)-CSR(NN)*DU(NN+1)+CSI(NN)*DV(NN+1)
      DV(NN)=DSI(NN)-CSI(NN)*DU(NN+1)-CSR(NN)*DV(NN+1)
3     CONTINUE
      IF(NDIF.EQ.1) GO TO 2004
      DO 2005 I=1,NX1
      U(I,1,2)=DU(I)
2005  V(I,1,2)=DV(I)
```

```
      GO TO 2006
2004  CONTINUE
      U(1,1,2)=.666666*DU(1)+.333333*DU(2)
      V(1,1,2)=.666666*DV(1)+.333333*DV(2)
      DO 33 I=2,NX1
      U(I,1,2)=.1666666*DU(I-1)+.666666*DU(I)+.1666666*DU(I+1)
33    V(I,1,2)=.1666666*DV(I-1)+.666666*DV(I)+.166666*DV(I+1)
2006  CONTINUE

C*************

C              THIS IS REGION ***II*** J=2,NY1  ,  I=1
      DO 20 J=2,NY1
      D1R=ARY*U(1,J-1,1)+BRY*U(1,J,1)+CRY*U(1,J+1,1)
     1-AIY*V(1,J-1,1)-BIY*V(1,J,1)-CIY*V(1,J+1,1)
      D1I=ARY*V(1,J-1,1)+BRY*V(1,J,1)+CRY*V(1,J+1,1)
     1+AIY*U(1,J-1,1)+BIY*U(1,J,1)+CIY*U(1,J+1,1)
      DSR(1)=(D1R*BRX+D1I*BIX)*BRIX
      DSI(1)=(D1I*BRX-D1R*BIX)*BRIX

C*************

C              THIS IS REGION ***I****    J=2,NY1,   I=2,NX1

      DO 21 I=1,NX11
      P=BRX-ARX*CSR(I)+AIX*CSI(I)
      Q=BIX-AIX*CSR(I)-ARX*CSI(I)
      PQ=1./(P*P+Q*Q)
      DIP1R=ARY*U(I+1,J-1,1)+BRY*U(I+1,J,1)+CRY*U(I+1,J+1,1)
     1-AIY*V(I+1,J-1,1)-BIY*V(I+1,J,1)-CIY*V(I+1,J+1,1)
      DIP1I=ARY*V(I+1,J-1,1)+BRY*V(I+1,J,1)+CRY*V(I+1,J+1,1)
     1+AIY*U(I+1,J-1,1)+BIY*U(I+1,J,1)+CIY*U(I+1,J+1,1)
      R=DIP1R+AIX*DSI(I)-ARX*DSR(I)
      S=DIP1I-AIX*DSR(I)-ARX*DSI(I)
      CSR(I+1)=(CRX*P+CIX*Q)*PQ
      CSI(I+1)=(CIX*P-CRX*Q)*PQ
      DSR(I+1)=(R*P+S*Q)*PQ
      DSI(I+1)=(S*P-R*Q)*PQ
21    CONTINUE

C*************

C              THIS IS REGION ***III***   J=2,NY1,   I=NX

      DNXR=ARY*U(NX,J-1,1)+BRY*U(NX,J,1)+CRY*U(NX,J+1,1)
     1-AIY*V(NX,J-1,1)-BIY*V(NX,J,1)-CIY*V(NX,J+1,1)
      DNXI=ARY*V(NX,J-1,1)+BRY*V(NX,J,1)+CRY*V(NX,J+1,1)
     1+AIY*U(NX,J-1,1)+BIY*U(NX,J,1)+CIY*U(NX,J+1,1)
      P=REMNX+AIX*CSI(NX1)-ARX*CSR(NX1)
      Q=IMMNX-AIX*CSR(NX1)-ARX*CSI(NX1)
      PQ=1./(P*P+Q*Q)
      R=DNXR+AIX*DSI(NX1)-ARX*DSR(NX1)
      S=DNXI-AIX*DSR(NX1)-ARX*DSI(NX1)
         DU(NX)=(R*P+S*Q)*PQ
         DV(NX)=(S*P-R*Q)*PQ
      DO 30 I=1,NX1
```

UNCLASSIFIED

```
      NN=NX-1
      DU(NN)=DSR(NN)-CSR(NN)*DU(NN+1)+CSI(NN)*DV(NN+1)
      DV(NN)=DSI(NN)-CSI(NN)*DU(NN+1)-CSR(NN)*DV(NN+1)
30    CONTINUE
      IF(NDIF.EQ.1) GO TO 2007
      DO 2008 I=1,NX1
      U(I,J,2)=DU(I)
2008  V(I,J,2)=DV(I)
      GO TO 2009
2007  CONTINUE
      U(1,J,2)=.333333*DU(2)+.666666*DU(1)
      V(1,J,2)=.333333*DV(2)+.666666*DV(1)
      DO 34 I=2,NX1
      U(I,J,2)=.1666666*DU(I-1)+.666666*DU(I)+.1666666*DU(I+1)
      V(I,J,2)=.1666666*DV(I-1)+.666666*DV(I)+.1666666*DV(I+1)
34    CONTINUE
2009  CONTINUE
20    CONTINUE
C*************

C              THIS IS REGION ***VII***  J=NY,  I=1

      D1R=REMNY*U(1,NY,1)-IMMNY*V(1,NY,1)+ARY*U(1,NY-1,1)
     1-AIY*V(1,NY-1,1)
      D1I=IMMNY*U(1,NY,1)+REMNY*V(1,NY,1)+AIY*U(1,NY-1,1)
     1+ARY*V(1,NY-1,1)
      DSI(1)=(D1I*BRX-D1R*BIX)*BRIX
      DSR(1)=(D1R*BRX+D1I*BIX)*BRIX

C*************

C              THIS IS REGION ***V****   J=NY,   I=2,NX1

      DO 40 I=1,NX11
      P=BRX-ARX*CSR(I)+AIX*CSI(I)
      Q=BIX-AIX*CSR(I)-ARX*CSI(I)
      PQ=1./(P*P+Q*Q)
      DIP1R=REMNY*U(I+1,NY,1)-IMMNY*V(I+1,NY,1)+ARY*U(I+1,NY-1,1)
     1-AIY*V(I+1,NY-1,1)
      DIP1I=IMMNY*U(I+1,NY,1)+REMNY*V(I+1,NY,1)+AIY*U(I+1,NY-1,1)
     1+ARY*V(I+1,NY-1,1)
      R=DIP1R+AIX*DSI(I)-ARX*DSR(I)
      S=DIP1I-AIX*DSR(I)-ARX*DSI(I)
      CSR(I+1)=(CRX*P+CIX*Q)*PQ
      CSI(I+1)=(CIX*P-CRX*Q)*PQ
      DSR(I+1)=(R*P+S*Q)*PQ
      DSI(I+1)=(S*P-R*Q)*PQ
40    CONTINUE

C*************

C              THIS IS REGION ***VIII***   J=NY,   I=NX

      DNXR=REMNY*U(NX,NY,1)-IMMNY*V(NX,NY,1)+ARY*U(NX,NY1,1)
     1-AIY*V(NX,NY1,1)
      DNXI=IMMNY*U(NX,NY,1)+REMNY*V(NX,NY,1)+AIY*U(NX,NY1,1)
```

```
     1+ARY*V(NX,NY1,1)
      P=REMNX+AIX*CSI(NX1)-ARX*CSR(NX1)
      Q=IMMNX-AIX*CSR(NX1)-ARX*CSI(NX1)
      PG=1./(P*P+Q*Q)
      R=DNXR+AIX*DSI(NX1)-ARX*DSR(NX1)
      S=DNXI-AIX*DSR(NX1)-ARX*DSI(NX1)
          DU(NX)=(R*P+S*Q)*PG
          DV(NX)=(S*P-R*Q)*PG
      DO 50 I=1,NX1
      NN=NX-I
      DU(NN)=DSR(NN)-CSR(NN)*DU(NN+1)+CSI(NN)*DV(NN+1)
      DV(NN)=DSI(NN)-CSI(NN)*DU(NN+1)-CSR(NN)*DV(NN+1)
50    CONTINUE
      IF(NDIF.EQ.1) GO TO 2010
      DO 2011 I=1,NX1
      U(I,NY,2)=DU(I)
2011  V(I,NY,2)=DV(I)
      GO TO 2012
2010  CONTINUE
      U(1,NY,2)=.333333*DU(2)+.666666*DU(1)
      V(1,NY,2)=.333333*DV(2)+.666666*DV(1)
      DO 35 I=2,NX1
      U(I,NY,2)=.1666666*DU(I-1)+.666666*DU(I)+.16666666*DU(I+1)
      V(I,NY,2)=.1666666*DV(I-1)+.666666*DV(I)+.16666666*DV(I+1)
35    CONTINUE
2012  CONTINUE
      GO TO 1000
200   CONTINUE

C*******************CONSTANTS*****************

      IMMNX=-ERF(X,2)
      IMMNY=ERF(Y,2)
      IMMY1=ERF(Y,2)
      CRY=CRX=ARY=ARX=1./6.
      IF(NDIF.EQ.0) CRY=CRX=ARY=ARX=0.0
      BRY=BRX=2./3.
      IF(NDIF.EQ.0) BRY=BRX=1.0
      CIY=AIY=-HZ/(2.*HY*HY)
      CIX=AIX= HZ/(2.*HX*HX)
      BIY= HZ/(HY*HY)
      BIX=-HZ/(HX*HX)

C*********************************************

      IF(NDIF.EQ.0) GO TO 3003
      E(1)=1./3.
      DO 3001 J=1,NY
      DO 3000 I=1,NX
      DU(I)=U(I,J,1)
3000  DV(I)=V(I,J,1)
      CALL MXTRID(NX,DU)
      CALL MXTRID(NX,DV)
      DO 3002 I=1,NX
```

UNCLASSIFIED

```
      U(I,J,1)=DU(I)
3002  V(I,J,1)=DV(I)
3001  CONTINUE
3003  CONTINUE
C**************

C          THIS IS REGION ***VI***    I=1,    J=1

      D1R=BRX*U(1,1,1)+2.*CRX*U(2,1,1)-BIX*V(1,1,1)-2.*CIX*V(2,1,1)
      D1I=BIX*U(1,1,1)+2.*CIX*U(2,1,1)+BRX*V(1,1,1)+2.*CRX*V(2,1,1)
      RR=REMY1*REMY1+IMMY1*IMMY1
      CSR(1)=(CRY*REMY1+CIY*IMMY1)/RR
      CSI(1)=(CIY*REMY1-CRY*IMMY1)/RR
      DSR(1)=(D1R*REMY1+D1I*IMMY1)/RR
      DSI(1)=(D1I*REMY1-D1R*IMMY1)/RR

C**************

C          THIS IS REGION ***II***    I=1,    J=2,NY1

      DO 500 J=1,NY11
      P=BRY-ARY*CSR(J)+AIY*CSI(J)
      Q=BIY-ARY*CSI(J)-AIY*CSR(J)
      PQ=1./(P*P+Q*Q)
      DJP1R=BRX*U(1,J+1,1)-BIX*V(1,J+1,1)+2.*CRX*U(2,J+1,1)-2.*CIX*V(2,
     1J+1,1)
      DJP1I=BIX*U(1,J+1,1)+BRX*V(1,J+1,1)+2.*CIX*U(2,J+1,1)+2.*CRX*V(2,
     1J+1,1)
      R=DJP1R-ARY*DSR(J)+AIY*DSI(J)
      S=DJP1I-AIY*DSR(J)-ARY*DSI(J)
      CSR(J+1)=(CRY*P+CIY*Q)*PQ
      CSI(J+1)=(CIY*P-CRY*Q)*PQ
      DSR(J+1)=(R*P+S*Q)*PQ
      DSI(J+1)=(S*P-R*Q)*PQ
500   CONTINUE

C**************

C          THIS IS REGION ***VII***   I=1,    J=NY

      DNYR=BRX*U(1,NY,1)-BIX*V(1,NY,1)+2.*CRX*U(2,NY,1)-2.*CIX*V(2,NY,1)
      DNYI=BIX*U(1,NY,1)+BRX*V(1,NY,1)+2.*CIX*U(2,NY,1)+2.*CRX*V(2,NY,1)
      P=REMNY-ARY*CSR(NY1)+AIY*CSI(NY1)
      Q=IMMNY-ARY*CSI(NY1)-AIY*CSR(NY1)
      PQ=1./(P*P+Q*Q)
      R=DNYR-ARY*DSR(NY1)+AIY*DSI(NY1)
      S=DNYI-ARY*DSI(NY1)-AIY*DSR(NY1)
         DU(NY)=(R*P+S*Q)*PQ
         DV(NY)=(S*P-R*Q)*PQ
      DO 334 J=1,NY1
      NN=NY-J
      DU(NN)=DSR(NN)-CSR(NN)*DU(NN+1)+CSI(NN)*DV(NN+1)
      DV(NN)=DSI(NN)-CSI(NN)*DU(NN+1)-CSR(NN)*DV(NN+1)
334   CONTINUE

      IF(NDIF.EQ.1) GO TO 3004
```

```
      DO 3005 J=1,NY1
      U(1,J,2)=DU(J)
3005  V(1,J,2)=DV(J)
      GO TO 3006
3004  CONTINUE
      U(1,1,2)=.166666*DU(2)+.666666*DU(1)
      V(1,1,2)=.166666*DV(2)+.666666*DV(1)
      DO 36 J=1,NY1
      U(1,J,2)=.1666666*DU(J-1)+.666666*DU(J)+.1666666*DU(J+1)
      V(1,J,2)=.1666666*DV(J-1)+.666666*DV(J)+.1666666*DV(J+1)
36    CONTINUE
3006  CONTINUE
C**************

C          THIS IS REGION ***IV***    I=2,NX1,    J=1

      DO 4 I=2,NX1
      D1R=ARX*U(I-1,1,1)-AIX*V(I-1,1,1)
     1+BRX*U(I,1,1)-BIX*V(I,1,1)
     2+CRX*U(I+1,1,1)-CIX*V(I+1,1,1)
      D1I=AIX*U(I-1,1,1)+ARX*V(I-1,1,1)
     1+BIX*U(I,1,1)+BRX*V(I,1,1)
     2+CIX*U(I+1,1,1)+CRX*V(I+1,1,1)
      DSR(1)=(D1R*REMY1+D1I*IMMY1)/RR
      DSI(1)=(D1I*REMY1-D1R*IMMY1)/RR

C**************

C          THIS IS REGION ***I***    I=2,NX1,    J=2,NY1

      DO 5 J=1,NY11
      P=BRY-ARY*CSR(J)+AIY*CSI(J)
      Q=BIY-ARY*CSI(J)-AIY*CSR(J)
      PG=1./(P*P+Q*Q)
      DJP1R=ARX*U(I-1,J+1,1)-AIX*V(I-1,J+1,1)
     1+BRX*U(I,J+1,1)-BIX*V(I,J+1,1)
     2+CRX*U(I+1,J+1,1)-CIX*V(I+1,J+1,1)
      DJP1I=AIX*U(I-1,J+1,1)+ARX*V(I-1,J+1,1)
     1+BIX*U(I,J+1,1)+BRX*V(I,J+1,1)
     2+CIX*U(I+1,J+1,1)+CRX*V(I+1,J+1,1)
      R=DJP1R-ARY*DSR(J)+AIY*DSI(J)
      S=DJP1I-AIY*DSR(J)-ARY*DSI(J)
      CSR(J+1)=(CRY*P+CIY*Q)*PG
      CSI(J+1)=(CIY*P-CRY*Q)*PG
      DSR(J+1)=(R*P+S*Q)*PG
      DSI(J+1)=(S*P-R*Q)*PG
5     CONTINUE

C**************

C          THIS IS REGION ***V***    I=2,NX1,    J=NY

      DNYR=ARX*U(I-1,NY,1)-AIX*V(I-1,NY,1)
     1+BRX*U(I,NY,1)-BIX*V(I,NY,1)
     2+CRX*U(I+1,NY,1)-CIX*V(I+1,NY,1)
      DNYI=AIX*U(I-1,NY,1)+ARX*V(I-1,NY,1)
```

UNCLASSIFIED

```
     1+BIX*U(I,NY,1)+BRX*V(I,NY,1)
     2+CIX*U(I+1,NY,1)+CRX*V(I+1,NY,1)
      P=REMNY-ARY*CSR(NY1)+AIY*CSI(NY1)
      Q=IMMNY-ARY*CSI(NY1)-AIY*CSR(NY1)
      PQ=1./(P*P+Q*Q)
      R=DNYR-ARY*DSR(NY1)+AIY*DSI(NY1)
      S=DNYI-ARY*DSI(NY1)-AIY*DSR(NY1)
         DU(NY)=(R*P+S*Q)*PQ
         DV(NY)=(S*P-R*Q)*PQ
      DO 300 J=1,NY1
      NN=NY-J
      DU(NN)=DSR(NN)-CSR(NN)*DU(NN+1)+CSI(NN)*DV(NN+1)
      DV(NN)=DSI(NN)-CSI(NN)*DU(NN+1)-CSR(NN)*DV(NN+1)
300   CONTINUE
      IF(NDIF.EQ.1) GO TO 3007
      DO 3008 J=1,NY1
      U(I,J,2)=DU(J)
3008  V(I,J,2)=DV(J)
      GO TO 3009
3007  CONTINUE
      V(I,1,2)=.166666*DV(2)+.666666*DV(1)
      U(I,1,2)=.166666*DU(2)+.666666*DU(1)
      DO 37 J=1,NY1
      U(I,J,2)=.1666666*DU(J-1)+.666666*DU(J)+.1666666*DU(J+1)
      V(I,J,2)=.1666666*DV(J-1)+.666666*DV(J)+.1666666*DV(J+1)
37    CONTINUE
3009  CONTINUE
4     CONTINUE

C**************

C          THIS IS REGION ***IX***    I=NX,    J=1

      D1R=REMNX*U(NX,1,1)-IMMNX*V(NX,1,1)+ARX*U(NX1,1,1)-AIX*V(NX1,1,1)
      D1I=IMMNX*U(NX,1,1)+REMNX*V(NX,1,1)+AIX*U(NX1,1,1)+ARX*V(NX1,1,1)
      DSR(1)=(D1R*REMY1+D1I*IMMY1)/RR
      DSI(1)=(D1I*REMY1-D1R*IMMY1)/RR
C**************

C          THIS IS REGION ***III***I=NX,    J=2,NY1

      DO 60 J=1,NY11
      P=BRY-ARY*CSR(J)+AIY*CSI(J)
      Q=BIY-ARY*CSI(J)-AIY*CSR(J)
      PQ=1./(P*P+Q*Q)
      DJP1R=REMNX*U(NX,J+1,1)-IMMNX*V(NX,J+1,1)+ARX*U(NX1,J+1,1)
     1-AIX*V(NX1,J+1,1)
      DJP1I=REMNX*V(NX,J+1,1)+IMMNX*U(NX,J+1,1)+ARX*V(NX1,J+1,1)
     1+AIX*U(NX1,J+1,1)
      R=DJP1R-ARY*DSR(J)+AIY*DSI(J)
      S=DJP1I-AIY*DSR(J)-ARY*DSI(J)
      CSR(J+1)=(CRY*P+CIY*Q)*PQ
      CSI(J+1)=(CIY*P-CRY*Q)*PQ
      DSR(J+1)=(R*P+S*Q)*PQ
      DSI(J+1)=(S*P-R*Q)*PQ
60    CONTINUE
```

```
C**************

C          THIS IS REGION ***VIII***    I=NX,    J=NY

      DNYR=REMNX*U(NX,NY,1)-IMMNX*V(NX,NY,1)+ARX*U(NX1,NY,1)
     1-AIX*V(NX1,NY,1)
      DNYI=REMNX*V(NX,NY,1)+IMMNX*U(NX,NY,1)+ARX*V(NX1,NY,1)
     1+AIX*U(NX1,NY,1)
      P=REMNY-ARY*CSR(NY1)+AIY*CSI(NY1)
      Q=IMMNY-ARY*CSI(NY1)-AIY*CSR(NY1)
      PQ=1./(P*P+Q*Q)
      R=DNYR-ARY*DSR(NY1)+AIY*DSI(NY1)
      S=DNYI-ARY*DSI(NY1)-AIY*DSR(NY1)
         DV(NY)=(S*P-R*Q)*PQ
         DU(NY)=(R*P+S*Q)*PQ
      DO 70 J=1,NY1
      NN=NY-J
      DU(NN)=DSR(NN)-CSR(NN)*DU(NN+1)+CSI(NN)*DV(NN+1)
      DV(NN)=DSI(NN)-CSI(NN)*DU(NN+1)-CSR(NN)*DV(NN+1)
70    CONTINUE
      IF(NDIF.EQ.1) GO TO 3010
      DO 3011 J=1,NY1
      U(NX,J,2)=DU(J)
3011  V(NX,J,2)=DV(J)
      GO TO 3012
3010  CONTINUE
      U(NX,1,2)=.166666*DU(2)+.666666*DU(1)
      V(NX,1,2)=.166666*DV(2)+.666666*DV(1)
      DO 38 J=1,NY1
      U(NX,J,2)=.1666666*DU(J-1)+.666666*DU(J)+.1666666*DU(J+1)
      V(NX,J,2)=.1666666*DV(J-1)+.666666*DV(J)+.1666666*DV(J+1)
38    CONTINUE
3012  CONTINUE
1000  CONTINUE
      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE PHASE(N,JJ)
      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QGMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOLT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SLM1,TAL,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ

      BANK,(0), /1/

      CALL INDEX(N)
      DO 1 I=1,NX
      DO 1 J=1,NY
      TEMPU=U(I,J,N)
      TEMPV=V(I,J,N)
      U(I,J,N)=TEMPU*COSF(CC(I,J))-TEMPV*SINF(CC(I,J))
1     V(I,J,N)=TEMPV*COSF(CC(I,J))+TEMPU*SINF(CC(I,J))
      IF(N.NE.2) GO TO 100
      KIMAX=1
      CIMAX=0.
      DO 2 J=1,NY
      IF(QI(1,J).LT.CIMAX) GO TO 2
      CIMAX=QI(1,J)
      KIMAX=J
2     CONTINUE
      CCMAX=ABSF(CC(1,KIMAX+5))
      PRINT 3,CCMAX,Z
3     FORMAT(* MAXIMUM PHASE CHANGE=*,E12.5,* RADIANS,AT Z=*,E12.5,*KM*)
      ZF=0.9*FL
      IF(Z.LT.ZF) GO TO 100
      IF(CCMAX.LT.0.5) GO TO 100
      HZ=HZ/2.
      NSTEP=NSTEP+2*(NSTEP-JJ)/2
      NNPLOT=NSTEP/(NPLOT-1)-1
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INITL (N)

      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECFNC,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IO,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOLT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ

      BANK,(0): /1/

      X=-HX
      DO 11 I=1,NX1
      X=X+HX
      Y=-(NY2-1)*HY
      DO 11 J=2,NY1
      Y=Y+HY
      IF((I-1)*(I-1)+(J-NY2)*(J-NY2).GT.100000) GO TO 100
      V(I,J,1)=VACAMP(X,Y,2)
      U(I,J,1)=VACAMP(X,Y,1)
      GO TO 200
100   U(I,J,1)=0.
      V(I,J,1)=0.
200   CONTINUE
11    CONTINUE
      CALL BOUNDS(1)
      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE GRAPH1(NFOCUS)

C     GRAPH1 PLOTS EQUI-INTENSITY CONTOURS AND BEAM PROFILE DURING ZLOOP


      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOLT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH2O,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SLM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON/XY/DEF

      DIMENSION IMAGE(61,61),QC(61),XQ(61)

      BANK,(0), /1/

C      PLOT EQUI-INTENSITY CONTOURS IN X,Y SPACE
      FOCUS=SQRTF(D)

      FF=FOCUS
      IF(NFOCUS.EQ.1) FF=1
      DEFL=DEF*WIDTH*FOCUS
      QNX=NX
      QNY=NY
      QQ=QNX/QNY
      XIN=8.0*FF*QQ
      YIN=8.0*FF

      YMIN=DEFL-(NY2-1)*HY*WIDTH*FOCUS/FF
      XMIN=0.0
      H1Y=NY*HY*WIDTH*FOCUS/(10.0*FF)
      H1X=NX*HX*WIDTH*FOCUS/(10.0*FF)
      CALL TOPOGRAF(CI,61,61,NX ,NY,1.,0.10,XIN,YIN,IMAGE,XMIN,H1X,
     1 4HF9.1,2HX ,-2,YMIN,H1Y,4HF9.1,3H Y ,3,Z ,PLTCAL,NXY,LPLOT,POWER,
     2 FL,D ,WIDTH,OMEGA,PH2O,VZERO,FF)

      NXY=NXY+1
      PLTCALL =0.0

      IF(LPLOT .EQ. 0)  GO TO 3

C      GET PROFILE ALONG WIND THROUGH CENTER OF BEAM AND PLOT

      DO 2 J=1,NY
      QQ(J)=QI(1,J )*POWER*EXPF(-ALPHA*WN*W2*ZETA)/(W2*D)
2     XQ(J)=(J-NY2)*4.0* FF  /(NY2-1)*4.0

      CALL SCALE(QQ,NY,6.0 ,YMIN,DY,1,TKY)
      XMIN=-(NY2-1)*HY*WIDTH*FOCUS/FF*DEFL
      DX=ABSF(XMIN-DEFL)/5.
      TKX=0.8
```

```
      CALL LINE(XQ,GQ,NY,1,-1,0,035,0)
      CALL AXIS(0,0,19HDISTANCE ALONG WIND,-19,8,0 ,0,0,TKX,
     1XMIN,DX,4HF9,1)
      CALL AXIS(0,0,24HINTENSITY AT BEAM CENTER,24,6,0 ,90,0,TKY
     1,YMIN,DY,4HF9,1)
      CALL PLOT(10,5,0,-3)

C     RE-INITIALIZE COUNTER FOR PLOTTING
3     KJ=0

      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE GRAPH2
C     GRAPH2 DOES FINAL PLOTTING AFTER Z-ITERATIONS ARE COMPLETED

      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ

      INTEGER XFORM,YFORM

      BANK,(0), /1/

C      PLOT MAXIMUM INTENSITY VS RANGE

      IGQ=IQ-1
      QQMAX(IQ)=0.0
      CALL SCALE(QQMAX,IG   , 7.0,YMIN,DY,1,TKY)
      CALL SCALE(ZM,IQ   ,8.0 ,XMIN,DX,1,TKX)
      XFORM=IFORMAT (XMIN,DX,8. ,TKX)
      YFORM=IFORMAT (YMIN,DY,7.,TKY)
      CALL LINE(ZM,QQMAX,IGQ,1,-1,.035,0)
      CALL AXIS(0.0,24HDISTANCE FROM LASER FACE,-24,8.0 ,0.0,TKX,
     1 XMIN,DX,XFORM)
      CALL AXIS(0.0,17HMAXIMUM INTENSITY,17, 7.0,90.0,TKY,YMIN,DY,
     1 YFORM)
      CALL PLOT(10.5,0.-3)

C      PLOT DEFLECTION OF MAX INTENSITY POINT VS RANGE
C      SKIP THIS PLOT IF BEAM STAYS AT ORIGIN

      IF (BETA.EQ.0.0) GO TO 2
      CALL SCALE(QJMAX,IG, 7.0,YMIN,DY,1,TKY)
      YFORM=IFORMAT (YMIN,DY,7.,TKY)
      CALL LINE(ZM,QJMAX,IGQ,1,-1,.035,0)
      CALL AXIS(0.0,24HDISTANCE FROM LASER FACE,-24,8.0 ,0.0,TKX,
     1 XMIN,DX,XFORM)
      CALL AXIS(0.0,15HBEAM DEFLECTION,15, 7.0,90.0,TKY,YMIN,DY,YFORM)
      CALL PLOT(10.5,0.-3)
2      CONTINUE

C      PLOT RATIO OF INTENSITY TO VAC,INT,--VS RANGE

      CALL SCALE(ZI,IGQ,8. ,XMIN,DX,1,TKX)
      CALL SCALE(PLTRAT(1),400+IG,7.0,YMIN,DY,1,TKY)
      XFORM=IFORMAT (XMIN,DX,8. ,TKX)
      YFORM=IFORMAT (YMIN,DY,7.,TKY)
      CALL LINE(ZI,PLTRAT(1),IGQ,1,0.,.08,1)
      CALL LINE(ZI,PLTRAT(201),IGQ,1,5.,.08,1)
```

```
 CALL LINE(ZI,PLTRAT(401),ICC,1,4,,08,1)
 CALL AXIS(0,0,24HDISTANCE FROM LASER FACE,-24,8,0 ,0,0,TKX,
1 XMIN,DX,XFORM)
 CALL AXIS(0,0,38HRATIO OF INTENSITY TO VACUUM INTENSITY,38,7,0,
190,0,TKY,YMIN,DY,YFORM)
 CALL SYMBOL(7,0,6,5,0,08,0,0,-1)
 CALL SYMBOL(7,25,6,4475,0,105,3H0,9,0,3)
 CALL SYMBOL(7,0,6,2,0,08,5,0,-1)
 CALL SYMBOL(7,25,6,1475,0,105,3H0,5,0,3)
 CALL SYMBOL(7,0,5,9,0,08,4,0,-1)
 CALL SYMBOL(7,25,5,8475,0,105,3H0,1,0,3)
 CALL PLOT(5,0 ,0,-3)

 RETURN
 END
```

UNCLASSIFIED

```
      FUNCTION IFORMAT (YMIN,DY,HEIGHT,TICK)

C     SELECT APPROPRIATE FORMAT FOR THIS DATA

      ENDTICK=YMIN+DY*INTF(HEIGHT/TICK)
      BIG=MAX1F(ABS(YMIN),ABS(ENDTICK))
      SMALL=DY
      CALL NORMAL (SMALL,IEXP)
      CALL NORMAL (BIG,NEXP)
   10 IF (IEXP.LT.-3) GO TO 14
      IF (NEXP.GE.4) GO TO 14
      IF (IEXP) 12,11,11

C     NO DECIMAL
  11  IDEC=0
      IRANGE=2+NEXP
      GO TO 13

C     WITH DECIMAL
  12  IDEC=-IEXP
      IRANGE=IDEC+3
      IF (NEXP.GE.0) IRANGE=IRANGE+NEXP

C     CONSTRUCT FORMAT
   13 IDEC=IDEC*8**8
      IRANGE=IRANGE*8**12
      IFORMAT=4HF0.0,OR,IDEC,OR,IRANGE
      RETURN
  14  IFORMAT=4HE8.1
      RETURN
      END
```

```
      SUBROUTINE NORMAL (ARG,IEXP)

C     NORMAL TAKES ANY NUMBER, ARG, AND NORMALIZES IT, IE, CONVERTS IT
C     TO THE FORM, ARG*10**IEXP,  WHERE 1.LE.ARG.LT.10.

      SIGN = +1.0
      IEXP = 0
      IF (ARG) 6,5,1
    6 SIGN = -1.0
      ARG = -ARG
    1 IF (ARG - 10.0) 2,4,4
    2 IF (ARG - 1.0) 3,5,5
    3 ARG = ARG*10.0
      IEXP = IEXP - 1
      GO TO 1
    4 ARG = ARG/10.0
      IEXP = IEXP + 1
      GO TO 1
    5 ARG = SIGN*ARG
      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE TOPOGRAF (T,NXD,NYD,NX,NY,F1,DELF,NC,XINCHES,YINCHES,
     1IMAGE,XMIN,DX,XFORMAT,XLABEL,NCX,YMIN,DY,YFORMAT,YLABEL,NCY,Z,PP,
     2NXY,LPLOT,POWER,FL,DL,WIDTH,OMEGA,PH20,VZERO,FF)

C     TOPOGRAF DRAWS A TOPOGRAPHICAL PLOT OF THE VALUES IN AN NX BY NY ARRAY, F,
C     DIMENSIONED F(NXD,NYX), USING SUBROUTINE CONTOUR.
C     CONTOURS WILL BE DRAWN FOR NC VALUES OF F, AT F1,F1-DELF,...,F1-(N-1)DELF
C     OR, IF DELF=0, THE ROUTINE WILL CALCULATE THE MAXIMUM AND MINIMUM VALUES
C     OF F AND DRAW NC CONTOURS BETWEEN THEM.
C     XINCHES = GRAPH LENGTH IN INCHES, YINCHES = GRAPH HEIGHT IN INCHES.
C     IMAGE = NX*NY STORAGE LOCATIONS FOR USE BY CONTOUR.
C     XMIN = 1ST X-VALUE, DX = X-VALUE INCREMENT, XFORMAT = FORMAT FOR X-VALUES.
C     XLABEL = NCX HOLLERITH CHARACTERS TO LABEL THE X-AXIS.
C     YMIN, DY, YFORMAT, YLABEL, NCY  PROVIDE CORRESPONDING VALUES FOR Y-AXIS.

      DIMENSION T(NXD,NYD)
      DIMENSION XOFF(6), YOFF(6)

      FOCUS=SQRTF(DL)
      XOFF(1)=XOFF(4)=2.5
      XOFF(2)=XOFF(5)=9.0
      XOFF(3)=XOFF(6)=15.5
      YOFF(1)=YOFF(2)=YOFF(3)=3.0
      YOFF(4)=YOFF(5)=YOFF(6)=-2.0
      XI=XINCHES
      YI=YINCHES
      DXT=DX
      DYT=DY
      IF (DELF.NE.0) GO TO 6

C     DETERMINE HIGHEST VALUE IN ARRAY FOR 1ST CONTOUR, AND DECREMENT,DF
C     TO GIVE DESIRED NUMBER OF CONTOURS BETWEEN HIGHEST AND LOWEST
C     VALUES, WHEN DELF IS NOT GIVEN

      FMIN = FMAX = T(1,1)
      DO 5 I=1,NX
      DO 5 J=1,NY
      IF (T(I,J) - FMIN) 2,5,3
    2 FMIN = T(I,J)
      GO TO 5
    3 IF (T(I,J) - FMAX) 5,5,4
    4 FMAX = T(I,J)
    5 CONTINUE

      DF = (FMAX - FMIN)/NC
      FLEVEL = FMAX
      GO TO 7

    6 FLEVEL = F1
      DF = DELF

C     DRAW NC CONTOURS, BEGINNING WITH THE FLEVEL VALUE

    7 DO 1 I=1,NC
      CALL CONTOUR (T,NXD,NYD,NX,NY,FLEVEL,XINCHES,YINCHES,IMAGE,NXY,
```

```
     1LPLOT)
   1 FLEVEL = FLEVEL * CF

      IF(LPLOT.EQ.1) GO TO 600
      X1=3.5+XOFF(NXY)
      X2=X1+0.3
      X3=X1+1.0
      Y1=2.0+YOFF(NXY)
      Y2=Y3=Y1
      GO TO 601
600   X1=1.0
      X2=1.3
      X3=2.0
      Y1=Y2=Y3=9.0
      EX=XI/10.
      EY=YI/10.
      CALL AXIS(0.0,YLABEL,NCY,YI,90.,EY,YMIN,DYT,YFORMAT)
      CALL AXIS(0.0,XLABEL,NCX,XI,0.0,EX,XMIN,DXT,XFORMAT)
601   CONTINUE
      IF(LPLOT.EQ.0) GO TO 603
      GO TO 602
603   CONTINUE
      YS=4.5+YOFF(NXY)
      XS=4.5+XOFF(NXY)
      CALL SYMBOL(XS,YS,.070,3,0.0,-1)
602   CONTINUE
      CALL SYMBOL( X1, Y1,.105,2HZ=,0.0,2)
      CALL NUMBER( X2, Y2,.105,Z,0.0,4HF7.4)
      CALL SYMBOL( X3, Y3,.105,2HKM,0.0,2)
      IF(PP.EQ.0.0) GO TO 20
      CALL SYMBOL(1.0,8.75,.105,6HPOWER=,0.0,6)
      CALL NUMBER(1.8,8.75,.105,POWER,0.0,5HF10.1)
      CALL SYMBOL(3.0,8.75,.105,5HWATTS,0.0,5)
      CALL SYMBOL(1.0,8.50,.105,2HF=,0.0,2)
      IF(FL.GT.1.0E+050) GO TO 110
      CALL NUMBER(1.3,8.5,.105,FL,0.0,4HF7.3)
      CALL SYMBOL(2.0,8.50,.105,2HKM,0.0,2)
      GO TO 210
110   CALL SYMBOL(1.3,8.50,.105,8HINFINITY,0.0,8)
210   CONTINUE
200   CONTINUE
      CALL SYMBOL(1.0,8.0,.105,6HWIDTH=,0.0,6)
      CALL NUMBER(1.8,8.0,.105,WIDTH,0.0,4HF7.3)
      CALL SYMBOL(2.8,8.0,.105,2HCM,0.0,2)
      IF(PP.EQ.2.0) GO TO 22
      CALL SYMBOL(1.0,7.5,.105,6HOMEGA=,0.0,6)
      CALL NUMBER(1.8,7.5,.105,OMEGA,0.0,4HF7.3)
      CALL SYMBOL(2.8,7.5,.105,7HRAD/SEC,0.0,7)
      CALL SYMBOL(1.0,7.25,.105,5HPH20=,0.0,5)
      CALL NUMBER(1.7,7.25,.105,PH20,0.0,4HF7.3)
      CALL SYMBOL(2.5,7.25,.105,4HTORR,0.0,4)
      CALL SYMBOL(1.0,7.0,.105,6HVZERO=,0.0,6)
      CALL NUMBER(1.7,7.0,.105,VZERO,0.0,4HF9.3)
      CALL SYMBOL(2.8,7.0,.105,6HCM/SEC,0.0,6)
      GO TO 20
22    CALL SYMBOL(1.0,7.5,.105,10HVACUUM RUN,0.0,10)
```

UNCLASSIFIED

```
20      CONTINUE
        IF(LPLOT.EQ.1) GO TO 700
        IF(NXY.EQ.6) GO TO 400
        XNEW=0.
        CALL PLOT (XNEW,0.,-3)
        GO TO 401
400     CALL PLOT(25.0,0.,-3)
        GO TO 401
700     CONTINUE
        P1=-4.0*FF    +4.0
        P2=+4.0*FF    +4.0
        CALL SYMBOL(P1,P1,.070,3,0.0,-1)
        CALL SYMBOL(P1,P2,.070,3,0.0,-1)
        CALL SYMBOL(P2,P2,.070,3,0.0,-1)
        CALL SYMBOL(P2,P1,.070,3,0.0,-1)
        CALL PLOT(10.5,0.,-3)
401     CONTINUE
        RETURN
        END
```

```
      SUBROUTINE CONTOUR (T,NXD,NYD,NX,NY,FLEVEL,XINCHES,YINCHES,IMAGE,
     1NXY,LPLOT)

      DIMENSION T(NXD,NYD), IMAGE(NXD,NYD)
      DIMENSION XOFF(100),YOFF(100)
      EQUIVALENCE (CXO, ICXO), (CYO,ICYO), (CX,ICX),(CY,ICY)

C     CONTOUR DOES INVERSE DOUBLE INTERPOLATION ON A 2-DIMENSIONAL ARRAY, F(X,Y)
C     WHEN CALLED WITH A GIVEN FLEVEL VALUE, IT RETURNS AFTER HAVING PLOTTED
C     A SET OF CONTOUR LINES, WHERE F = FLEVEL, ON A GRAPH XINCHES LONG,
C     AND YINCHES HIGH.

C     NXD AND NYD SPECIFY THE SIZE OF ARRAY T GIVEN IN THE DIMENSION
C     STATEMENT, WHILE NX AND NY DEFINE THE AMOUNT OF ARRAY T ACTUALLY
C     USED.

      XOFF(1)=XOFF(4)=2.5
      XOFF(2)=XOFF(5)=9.0
      XOFF(3)=XOFF(6)=15.5
      YOFF(1)=YOFF(2)=YOFF(3)=3.0
      YOFF(4)=YOFF(5)=YOFF(6)=-2.0
      JUP=0
      XFACTOR = XINCHES/(NX-1)
      YFACTOR = YINCHES/(NY-1)
      IF(LPLOT.NE.1) GO TO 400
      XOFF(NXY)=YOFF(NXY)=0.
400   CONTINUE
      QNX=NX
      QNY=NY
      QQ=QNY/QNX
      XSHIFT=XOFF(NXY)+4.0-XINCHES*QQ/2.0
      YSHIFT=YOFF(NXY)+4.0-YINCHES/2.0

C     LOAD IMAGE ARRAY

      DO 2 IY=1,NY
      DO 2 IX=1,NX
      IF (T(IX,IY).GE.FLEVEL) GO TO 1
      IMAGE(IX,IY) = -1
      GO TO 2
    1 IMAGE(IX,IY) = 1
    2 CONTINUE

C     SCAN IMAGE FOR THE 1ST POINT OF A REGION

      IYSTART = 1
    3 DO 4 IY=IYSTART,NY
      DO 4 IX=1,NX
      IF (IMAGE(IX,IY).EQ.1) GO TO 5
    4 CONTINUE
      RETURN

C     LIFT PEN AND BRING TO STARTING POINT, AND SKIRT THE REGION FOUND.

    5 IYSTART = IY
      IF (IY.EQ.1) GO TO 6
```

UNCLASSIFIED

```
      IF (IMAGE(IX,IY-1).EQ.0) GO TO 8
      CYO = (IY-1-(T(IX,IY)-FLEVEL)/(T(IX,IY)-T(IX,IY-1)))*YFACTOR
      GO TO 7
    6 CYO = 0
    7 CXO = (IX-1)*XFACTOR
      CXO=CXO+XSHIFT
      CYO=CYO+YSHIFT
      CALL PLOT (CXO,CYO,3)
      INOUT = 2
      GO TO 20

C     START AN INNER BOUNDARY
    8 INOUT = 1
      CXO = (IX-2)*XFACTOR
      CYO=(IY-2+(T(IX-1,IY-1)-FLEVEL)/(T(IX-1,IY-1)-T(IX-1,IY)))*YFACTOR
      CXO=CXO+XSHIFT
      CYO=CYO+YSHIFT
      CALL PLOT (CXO,CYO,3)
      GO TO 20
C     SKIRT DIRECTION IS ALWAYS COUNTER-CLOCKWISE FOR AN EXTERNAL BOUNDARY,
C     AND CLOCKWISE FOR AN INTERNAL BOUNDARY (IE, THE INSIDE OF THE REGION
C     IS ALWAYS TO THE LEFT OF THE SKIRT DIRECTION.
C     POSITIVE X-CROSSING
   10 CX = (IX-1)*XFACTOR
      IF(IY.EQ.NY) GO TO 11
      CY = (IY-1+(T(IX,IY)-FLEVEL)/(T(IX,IY) - T(IX,IY+1)))*YFACTOR
      GO TO 12
   11 CY = (NY-1)*YFACTOR
12    CONTINUE
      CY=CY+YSHIFT
      CX=CX+XSHIFT
      CALL PLOT(CX,CY,2)
      IF (CX.NE.CXO) GO TO 16
      YGAP = ABSF(CY - CYO)
      IF (YGAP.LT..001) GO TO 3

   16 IF (IX.EQ.NX) GO TO 40
      IF (T(IX+1,IY) - FLEVEL) 40,13,13
   13 IF (IY.EQ.NY) GO TO 14
      IF (T(IX+1,IY+1) - FLEVEL) 14,15,15
   14 IX = IX+1
      GO TO 10
   15 IX = IX+1
      IY = IY+1
      GO TO 20

C     POSITIVE Y-CROSSING

   20 IF (IX.EQ.1) GO TO 21
        CX = (IX-1-(T(IX,IY) - FLEVEL)/(T(IX,IY) - T(IX-1,IY)))*XFACTOR
      GO TO 22
   21 CX = 0
   22 CY = (IY-1)*YFACTOR
      CY=CY+YSHIFT
      CX=CX+XSHIFT
      CALL PLOT(CX,CY,2)
```

```
      DO 26 I=IX,NX
      IF (IMAGE(I,IY).LT.1) GO TO 28
   26 IMAGE(I,IY) = 0

   28 IF (IY.EQ.NY) GO TO 10
      IF (T(IX,IY+1) - FLEVEL) 10,23,23
   23 IF (IX.EQ.1) GO TO 24
      IF (T(IX-1,IY+1) - FLEVEL) 24,25,25
   24 IY = IY + 1
      GO TO 20
   25 IX = IX-1
      IY = IY+1
      GO TO 30
C     NEGATIVE X-CROSSING

   30 CX = (IX-1)*XFACTOR
      IF (IY.EQ.1) GO TO 31
      CY = (IY-1-(T(IX,IY) - FLEVEL)/(T(IX,IY) - T(IX,IY-1)))*YFACTOR
      GO TO 32
   31 CY = 0
32    CONTINUE
      CX=CX+XSHIFT
      CY=CY+YSHIFT
      CALL PLOT(CX,CY,2)

      IF (CX.NE.CX0) GO TO 33
      IF (CY.EQ.CY0) GO TO 3

   33 IF (IX.EQ.1) GO TO 20
      IF (T(IX-1,IY) - FLEVEL) 20,34,34
   34 IF (IY.EQ.1) GO TO 35
      IF (T(IX-1,IY-1) - FLEVEL) 35,36,36
   35 IX = IX-1
      GO TO 30
   36 IX = IX-1
      IY = IY-1
      GO TO 40

C     NEGATIVE Y-CROSSING

   40 CY = (IY-1)*YFACTOR
      IF (IX.EQ.NX) GO TO 41
      CX = (IX-1+(T(IX,IY) - FLEVEL)/(T(IX,IY) - T(IX+1,IY)))*XFACTOR
      GO TO 42
   41 CX = (NX-1)*XFACTOR
42    CONTINUE
      CX=CX+XSHIFT
      CY=CY+YSHIFT
      CALL PLOT(CX,CY,2)

      IF (IY.EQ.1) GO TO 30
      IF (T(IX,IY-1) - FLEVEL) 30,43,43
   43 IF (IX.EQ.NX) GO TO 44
      IF (T(IX+1,IY-1) - FLEVEL) 44,45,45
   44 IY = IY-1
      GO TO 40
   45 IX = IX+1
      IY = IY-1
      GO TO 10

      END
```

UNCLASSIFIED

```
      SUBROUTINE INDEX (KK)
C*********************************************************************
C       THIS ROUTINE INTEGRATES THE LOCAL INTENSITY FROM INFINITY UP
C       TO THE POINT X,Y ALONG THE Y (WIND) DIRECTION FOR A GAS WHICH IS
C       BEING HEATED AND COOLED BY THE LIGHT BEAM, AND MULTIPLIES THE
C       RESULT BY CONST TO GIVE THE INDEX CHANGE AT EACH  POINT IN
C       THE TRANSVERSE PLANE, THE RESULTS ARE STORED IN ARRAY CC(I,J)
C*********************************************************************
      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON/XY/DEF

      DIMENSION A(61),B(61)

      BANK,(0), /1/

      ARG=WIDTH*FOCUS/(TAU*(VZERO+OMEGA*ZETA*W2*WN))
      CONST=BETA*EXPF(-WN*W2*ALPHA*ZETA)/(1.+OMEGA*WN*W2*ZETA/VZERO)
      GO TO (3,4) KK
3     ZP=ZZ+HZ/2.
      GO TO 5
4     ZP=ZZ-HZ/2.
5     FHAT=F/(WN*W2)
      ZETB=(TANF(ZP-ATANF(WN*W2/F))+WN*W2/F)/(1.+WN*WN*W2*W2/(F*F))
      ZB=ZETB*WN*W2
      Z1B=1.-ZB/F
      DB=ZETB*ZETB+Z1B*Z1B
      FOCUSB=SQRTF(DB)
      CONSTB=BETA*EXPF(-WN*W2*ALPHA*ZETB)/(1.+OMEGA*WN*W2*ZETB/VZERO)
      CONST=.125*HZ*(CONST*FOCUS+CONSTB*FOCUSB)
      EX=EXPF(-ARG*HY)
      HDZ=HZ/4.
      DO 1 I=1,NX
      X=(I-1)*HX
      Y=-(NY2-1)*HY
      Y=Y+DEF
      CC(I,1)=-(X*X+Y*Y-2.)*HDZ
      UV2=U(I,2,KK)*U(I,2,KK)+V(I,2,KK)*V(I,2,KK)
      UV1=U(I,1,KK)*U(I,1,KK)+V(I,1,KK)*V(I,1,KK)
      A(2)=-CONST*HY*0.5*(UV1+UV2)
      B(2)=+CONST*DELTA*0.5*HY*(UV2+EXPF(-ARG*HY)*UV1)
      Y=-(NY2-2)*HY
      Y=Y+DEF
      CC(I,2)=A(2)+B(2)-(X*X+Y*Y-2.)*HDZ
```

```
      DO 1 J=3,NY1
      Y=-(NY2-J)*HY
      Y=Y+DEF
      UVJ=(U(I,J,KK)*U(I,J,KK)+V(I,J,KK)*V(I,J,KK))
      UVJ1=(U(I,J-1,KK)*U(I,J-1,KK)+V(I,J-1,KK)*V(I,J-1,KK))
      A(J)=A(J-1)-CONST*HY*0.5*(UVJ+UVJ1)
      B(J)=B(J-1)*EX+CONST*DELTA*0.5*HY*(UVJ+UVJ1*EX)
      CC(I,J)=A(J)+B(J)-(X*X+Y*Y-2.)*HDZ
1     CONTINUE

      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE INTENS

C           IGATE = 1 FOR 1ST CALL, BEFORE Z ITERATIONS
C           IGATE = 2 DURING Z-ITERATIONS, TO SKIP PLOTTING
C           IGATE = 3 DURING Z-ITERATIONS, TO DO PLOTTING


      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH2O,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SLM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON /REL/ CONMIN(9),AREA(9),SUMA(9)

      DATA (IGATE=1)

      BANK,(0), /1/

      GO TO (100,1) IGATE
C     THIS SECTION CALCULATES THE INTENSITY AT ALL MESH POINTS
C     IT IS DONE ONCE, AT FIRST INTENS CALL
  100 QMAX=0
      DO 102 J=1,NY
      DO 102 I=1,NX
      QI(I,J)=U(I,J,1)*U(I,J,1)+V(I,J,1)*V(I,J,1)
      IF(QI(I,J).GT.QMAX)  QMAX=QI(I,J)
102   CONTINUE
      DO 103 I=1,9
  103 CONMIN(I) = (10-I)/10.0
      GO TO 7


C     CALCULATE INTENSITIES (ALL SUBSEQUENT CALLS TRANSFER HERE)
    1 IF (KP.EQ.KKMAX.OR.KJ.EQ.NNPLOT) IGATE=3

C      FACTOR FOR SHRINKING COORDINATE SYSTEM
      F2=D
      DO 3 J=1,NY
      DO 3 I=1,NX
3     QI(I,J)  =U(I,J,2)*U(I,J,2)+V(I,J,2)*V(I,J,2)

      GO TO (7,7,4) IGATE

C     CALCULATE MAXIMUM INTENSITIES FOR PLOTTING
    4 QMAX=QCMAX=0
      DO 6 J=1,NY
      IF(QI(1,J).LE.QCMAX) GO TO 5
      QCMAX=QI(1,J)
      JCMAX=J
5     DO 6 I=1,NX
      IF(QI(I,J) .LE. QMAX) GO TO 6
```

```
      QMAX=QI(I,J)
      IMAX=I
      JMAX=J
    6 CONTINUE

C     CALCULATE THE TOTAL INTEGRATED INTENSITY

    7 SUM=0
      DO 8 I=1,9
    8 SUMA(I) = AREA(I) = 0

C     CALCULATE SUM FOR ENERGY CONSERVATION TEST AND OUTPUT
      DO 12 I=1,NX
      DO 12 J=1,NY
      TWO=2.0
      IF(I.EQ.1) TWO=1.0
      TEMP1=HX*HY
      TEMP2=POWER*QI(I,J)*TEMP1
      SUM=SUM+TEMP2*TWO
      GO TO (9,12,9) IGATE

C     PROCESS DATA FOR INTENSITY PLOTS
    9 TS=QI(I,J)/QMAX
      IF(TS .LT. 0.1)  GO TO 12
      DO 11 K=1,9
      IF(TS .LT. CONMIN(K))  GO TO 11
      DO 10 L=K,9
      SUMA(L)=SUMA(L)+TEMP2*TWO
   10 AREA(L)=AREA(L)+TEMP1*TWO
      GO TO 12
   11 CONTINUE
   12 CONTINUE

C     SKIP THIS SECTION UNLESS DOING PLOTS
      GO TO (13,24,14) IGATE

   13 SUM1 = SUM
      SUMAF = F2 = ZW2WNF = 1.0
      II=1
      ZI(1) = 0
      GO TO 15
14    CONTINUE
      SUMAF=EXPF(-ALPHA*WN*W2*ZETA)
      II = IO+1
      ZI(II)=Z
      PRINT 160
160   FORMAT(* *****************************************************
     1*****************************************************)
      PRINT 161,Z
161   FORMAT(//* RANGE=*,F10.3,* KILOMETERS*//)
   15 PRINT 16
   16 FORMAT(1H0*CONMIN*7X*AVGINT*11X*AREA*11X*VACINT*11X*VAREA*8X
     1*RATIO-I*6X*RATIO-A*)

      DO 20 I=1,9
      SUMA(I) = SUMA(I)*SUMAF
```

UNCLASSIFIED

```
      AREA(I)=AREA(I)*W2*F2
      AVGINT=SUMA(I)/AREA(I)
      VACINT=POWER*(1.-CONMIN(I))/(PI*W2*LOGF(1./CONMIN(I))*D)
     1*EXPF(-ALPHA*WN*W2*ZETA)
      VAREA = PI*W2*LOGF(1.0/CONMIN(I))*D
      RATIOI=AVGINT/VACINT
      RATIOA=AREA(I)/VAREA
      PRINT 17, CONMIN(I),AVGINT,AREA(I),VACINT,VAREA,RATIOI,RATIOA
   17 FORMAT(1X,F6.4,4E16.5,2F13.6)
      IF (I.NE.1) GO TO 18
      PLTRAT(II,1) = RATIOI
      GO TO 20
   18 IF (I.NE.5) GO TO 19
      PLTRAT(II,2) = RATIOI
      GO TO 20
   19 IF (I.NE.9) GO TO 20
      PLTRAT(II,3) = RATIOI
   20 CONTINUE

      PRINT 21
   21 FORMAT (/)

      GO TO (22,24,25) IGATE
22    CONTINUE
      PRINT 23, SUM
   23 FORMAT(30H INITIAL INTEGRATED INTENSITY=E15.8,6H WATTS/)
      GO TO 25

24    CONTINUE

   25 IGATE = 2
      RETURN
      END
```

```
      SUBROUTINE OUTPUT (II,JJ,KK)

C     II DENOTES THE TYPE OF OUTPUT TO BE GENERATED BY THIS SUBROUTINE
C        II=1 GIVES AMPLITUDE PRINTOUT (ARRAY U,V)
C        II=2 STORES PLOTTING DATA AT ALTERNATE JJ STEPS, AND GIVES PEAK
C             INTENSITY, CENTRAL BEAM DEFLECTION, GSN, AND RELI PRINTOUT
C             EVERY KKMAX-TH CALL,
C        II=3 GIVES DENSITY CHANGE, PHASE, AND AMPLITUDE PRINTOUT,
C        II=4 GIVES NORMALIZED INTENSITY PRINTOUT AFTER =INTENSITY PLOT
C     JJ IS THE MAIN PROGRAM Z-ITERATION INDEX
C     KK IS THE LAST ARRAY COLUMN IN AMPLITUDE PRINTOUT, U,V(I,J,KK),
C        KK SHOULD =1 OR 2 FOR II=1, AND =3 FOR II=3,


      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SLM1,TAL,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON /XY/ DEF
      COMMON /PEAKS/ QQGMAX,RUNPP,ZPP

C     DIMENSIONED ARRAYS
C        ARCT       HOLDS PHASE INFORMATION
C        DENS       ARRAY CONTAINING DENSITY CHANGES
C        NQI        STORES A ROW OF NORMALIZED INTENSITIES FOR PRINTOUT

      DIMENSION ARCT(61),DENS(61,61),NQI(61)

      BANK,(0), /1/

      NX3=36
      GO TO (10,20,30,40), II


C      PRINT AMPLITUDES

   10 J1=1
11    PRINT 12,Z
   12 FORMAT(8H1U AT Z=E12,5)
      PRINT FORM2,((U(I,J,KK),I=1,NX3,2),J=J1,NY)
      PRINT 13,Z
   13 FORMAT(8H1V AT Z=E12,5)
      PRINT FORM2,((V(I,J,KK),I=1,NX3,2),J=J1,NY)
      AZ=WIDTH*FOCUS*HY
      PRINT 49,Z,AZ
49    FORMAT(*0INTENSITY VS R AT Z=*F10,3* AT INTERVALS OF*F10,3*CM*)
      PRINT 50,(QI(1,J),J=NY2,NY)
50    FORMAT(7(1X,F15,11))
      IF(II,NE,1) GO TO 28
```

UNCLASSIFIED

```
      DO 17 N=1,2
      Y=-NY2*HY
      DO 14 K=1,NY
      Y=Y+HY
      X=-HX
      DO 14 J=1,NX
      X=X+HX
14    DENS(J,K)=VACAMP(X,Y,N)
      GO TO (18,19) N
18    CONTINUE
      PRINT 15,Z
15    FORMAT(*1 UANALYTIC AT Z=*E12,5)
      GO TO 2000
19    CONTINUE
      PRINT 16,Z
16    FORMAT(*1 V ANALYTIC AT Z=*E12,5)
2000  CONTINUE
      PRINT FORM2,((DENS(J,K),J=1,NX3,2),K=1,NY)
17    CONTINUE
      RETURN

C     STORE PLOTTING DATA AT ALTERNATE STEPS (JJ ODD)

20    CONTINUE
C     CALCULATE AND PLOT MAXIMUM INTENSITY EVERY KKMAX STEPS

   21 IF(KP.EQ.KKMAX.OR.KJ.EQ.NNPLOT) GO TO 22
      GO TO 28
   22 KP=0
      PRINT 23,SUM
23    FORMAT(* INTEGRATED INTENSITY =*,E15.8,* WATTS*)
      SUM2 = ABSF((SUM-SUM1)/SUM1)
      PRINT 24, SUM2
   24 FORMAT(1X,*DELTA-E/E=*E12.5)

C     MAXIMUM INTENSITY AND BEAM DEFLECTION CALCULATION

      QM=QI(IMAX,JMAX)-QI(IMAX,JMAX-1)
      QP=QI(IMAX,JMAX)-QI(IMAX,JMAX+1)
      QQMAX(IQ)=QI(IMAX,JMAX)+0.125*(QP-QM)**2/(QP+QM)
      QQMAX(IQ)=QQMAX(IQ)*POWER*EXPF(-ALPHA*WN*W2*ZETA)/(W2*D)
      ZM(IQ)=Z
      PRINT 25,QQMAX(IQ)
25    FORMAT(* PEAK INTENSITY=*,E12.5,* WATTS/SQ-CM*)
      QP=QI(1,JCMAX)-QI(1,JCMAX+1)
      QM=QI(1,JCMAX)-QI(1,JCMAX-1)
      SJMAX=JCMAX+(QM-QP)/(2.*(QM+QP))
      QJMAX(IQ)=((SJMAX-1.)*HY-(NY2-1)*HY)*SQRT(D)*WIDTH*DEF*WIDTH*FOCUS
      QQQMAX=QI(1,JCMAX )+0.125*(QP-QM)**2/(QP+QM)
      QQQMAX=QQQMAX*POWER*EXPF(-ALPHA*WN*W2*ZETA)/(W2*D)

C     RUNPP WILL STORE PEAK POWER FOR RUN
      IF(JJ .EQ. 1) RUNPP=QQQMAX
      IF(QQQMAX .LE. RUNPP)  GO TO 27
      RUNPP=QQQMAX
```

```
      ZPP=Z
27    CONTINUE
      PRINT 26,QQQMAX
26    FORMAT(* CENTRAL PEAK INTENSITY=*,E12.5,* WATTS/SQ-CM*)

      PRINT 160
160   FORMAT(* +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
     1-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-*)
      IQ=IQ+1
   28 RETURN
30    ZZM=Z
C      PRINT AMPLITUDES AND RETURN
      J1=1
      GO TO 11


C     NORMALIZE ALL INTENSITIES TO MAX ON A SCALE OF 0 TO 10, AND PRINT

40    PRINT 41,Z
   41 FORMAT (16H1INTENSITY AT Z= E12.5)
      PRINT 42
   42 FORMAT (12H X=-50,Y=-50, 90X,9HX=0,Y=-50 /)

      DO 44 J=1,NY
      DO 43 I=1,40
   43 NQI(I) = XFIXF(10.*QI(I,J)/QMAX)
   44 PRINT FORM1, (NQI(I),I=1,40)

      PRINT 45
   45 FORMAT (12H X=-50,Y=+50, 90X,9HX=0,Y=+50 /)
      IF (NOUT.EQ.1) PRINT 46,Z
   46 FORMAT (38H ENERGY CHANGES TOO LARGE, ABORT AT Z= E12.5,3H KM //)

      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE FOLLOW(DEFL)
      COMMON /1/ CC(61,61),QI(61,61),U(61,61,2),V(61,61,2)
      COMMON /2/ FORM1(2),FORM2(2),PLTRAT(200,3),QJMAX(200),QQMAX(200),
     1 ZI(200),ZM(200)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOLT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PHZO,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON/XY/DEF

      BANK,(0), /1/

      JCENM=NY2
      CMAX=0.
      DO 1 J=1,NY
      IF(QI(1,J).LE.CMAX) GO TO 1
      CMAX=QI(1,J)
      JCENM=J
1      CONTINUE
      IF(JCENM.GE.NY2+1) GO TO 10
      IF(JCENM.LE.NY2-1) GO TO 11
      GO TO 12
10    CONTINUE
      DO 100 I=1,NX
      DO 100 J=1,NY1
      QI(I,J)=QI(I,J+1)
      CC (I,J)=CC (I,J+1)
      DO 100 K=1,2
      U(I,J,K)=U(I,J+1,K)
      V(I,J,K)=V(I,J+1,K)
100   CONTINUE
      JMAX=JMAX-1
      JCMAX=JCMAX-1
      GO TO 13
11    CONTINUE
      DO 200 I=1,NX
      DO 200 J=1,NY1
      JJ=NY+1-J
      QI(I,JJ)=QI(I,JJ-1)
      CC (I,JJ)=CC (I,JJ-1)
      DO 200 K=1,2
      U(I,JJ,K)=U(I,JJ-1,K)
      V(I,JJ,K)=V(I,JJ-1,K)
200   CONTINUE
      JMAX=JMAX+1
      JCMAX=JCMAX+1
      GO TO 15
13    CONTINUE
      DEFL=DEFL+HY*WIDTH*FOCLS
      GO TO 14
15    CONTINUE
      DEFL=DEFL-HY*WIDTH*FOCLS
14    CONTINUE
```

```
      PRINT 400,DEFL,Z
400   FORMAT(* DEFLECTION=*,E12.5,*CM AT Z =*,F10.3,* KM*)
12    CONTINUE
      DEF=DEFL/(WIDTH*FOCUS)
      CALL BOUNDS(2)
      RETURN
      END



      SUBROUTINE MXTRID(N,D)
      COMMON/ABC/A(61),B(61),C(61)
      DIMENSION D(61),CS(61),DS(61)
      CS(1)=C(1)/B(1)
      DS(1)=D(1)/B(1)
      DO 1 K=2,N
      BB=B(K)-A(K-1)*CS(K-1)
      CS(K)=C(K)/BB
1     DS(K)=(D(K)-A(K-1)*DS(K-1))/BB
      D(N)=DS(N)
      N1=N-1
      DO 2 L=1,N1
      K=N-L
2     D(K)=DS(K)-CS(K)*D(K+1)
      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE FINALS

C     PRINT INPUT DATA AND SELECTED OUTPUT (IN DUPLICATE)
      COMMON /SINGLS/ AAA,ALFCON,ALFSUM,ALPHA,BBB,BETA,BETAZ,C,CZERO,D,
     1 DDD,DELTA,DIAM,ECHNG,EEE,F,FL,FOCUS,GAMMA,HX,HY,HZ,HZV,HZX,HZY,
     2 IMAX,IQ,IT,JCMAX,JMAX,     KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,
     3 NCONS,NDEL,NDIF,NNPLOT,NOUT,NPLOT,NSTEP,NTAU,NX,NX1,NX2,
     4 NXY,NY,NY1,NY2,OMEGA,P,PH20,PI,PLTCAL,POWER,QMAX,REFRAC,RO,ROZERO
     5 ,STARTT,STOPP,SUM,SUM1,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,
     6 ZT,ZZ
      COMMON/NNN/NFOCUS
      COMMON/XY/DEF
      COMMON /TYPE/ CASE(4)
      COMMON /PEAKS/ QQQMAX,RUNPP,ZPP
      COMMON /REL/ CONMIN(9),AREA(9),SUMA(9)
      PRINT 28
28    FORMAT(1H1)
      DO 66  M=1,2
      PRINT 30,CASE,DIAM
30    FORMAT(2X***60X,2A8,7X***/15X,48X,2A8//15X,9X*INPUT DATA:*25X
     1  *DIAM*F6* CM*)
      TS1=F/100./1852.
      TS2=POWER/1000.
      PRINT 32,FL,TS1,HX,NX,TS2
32    FORMAT(15X,45X*FOCUS*F5.1* KM*/15X,3X*CARD 1*14X*CARD 2*16X,F11.2,
     1  * NAUT MI*/15X*HX*F7.2,11X*NX*I7,16X*POWER*F9.2* KW*)
      TS1=(VZERO/100./1852.)*3600.
      PRINT 34,HY,NY,VZERO,HZV,NSTEP,TS1
34    FORMAT(15X*HY*F7.2,11X*NY*I7,16X*WIND*F7* CM/SEC*/15X*HZ*F11.6,7X,
     1  *NSTEP*I4,16X,F11.1,* KTS*)
      TS1=OMEGA*1000.
      TS2=ALPHA*100000.
      PRINT 36,DIAM,NPLOT,TS1,WN,KQMAX,TS2
36    FORMAT(15X*DIAM*F4* CM*9X*NPLOT*I3,17X*SLUING*F4* MRAD/SEC*/15X,*W
     1N*F10.2*/CM*5X*KQMAX*I4,16X*ALPHA*F6.2* /KM*)
      TS1= *10.*ALOG10(EXPF(-ALPHA*100000.))
      PRINT 38,F,STARTT,TS1,ZT,STOPP,LPLOT,NFOCUS,NDIF
38    FORMAT (15X*F*F10* CM*6X*STARTT*F5.2* KM*11X*LOSS*F7.2* DB/KM*/
     1  15X*ZT*F7.2* KM*8X*STOPP*F6.2* KM*/15X,20X*LPLOT*I3/15X,20X
     2  *NFOCUS*I2/35X*NDIF*I4)
      TS1=QQQMAX/1000.
      PRINT 40,TS1,GAMMA,REFRAC,PH20,CZERO,P
40    FORMAT(15X,3X*CARD 3*36X*ICP(F)*F8.3*KW/CM2*/15X*GAMMA*F5.1,13X
     1  *CARD 4*/15X*REFRAC*F6.3*CM3/GM*2X*PH20*F7.1* TORR*/15X*C*F11
     2  *CM/SEC*2X*P*F8.1X*TORR*)
      TS1=THETA*(180./PI)
      PRINT 42,VZERO,OMEGA,POWER,ECHNG,TS1,ROZERO
42    FORMAT(15X*VZERO*F6*CM/SEC*3X*OMEGA*F6.3* RAD/SEC*/15X*POWER*F8
     1  * W*5X*ECHNG*F6.2,14X*DIMENSIONLESS*/15X*THETA*F5.1*DEG*7X*RO*
     2  F11.5*GM/CM3*15X*PARAMETERS:*)
      PRINT 43,TEMP,BETAZ
43    FORMAT(35X,*TEMP *F5.1* DEG CENT*/66X*BETA*F9.2)
      TS1=OMEGA*F/VZERO
      TS2=ALPHA*F
      PRINT 44, TS1,NDEL,CASE(1),CASE(2),TS2
44    FORMAT(15X,3X*CARD 5*14X*CARD 6*16X,10HOMEGA*F/VO,F9.5/
```

```
     1  15X*NDEL*I4,12X*CASE: *2A8,6X,7HALPHA*F,F9.6)
      TS1=WIDTH/F
      TS2=WN*WIDTH
      TS3=BETAZ*F/WIDTH
      PRINT 46,DELTA,CASE(3),CASE(4),TS1,NTAU,TS2,TAU,TS3
46    FORMAT(15X*DELTA*F7.3,8X,2A8,16X*A/F*F10.8/15X*NTAU*I4,44X,
     1  3HK*A,F9.2/15X*TAU*F9.3* SEC*7X*RUN QUALITY WAS*9X,8HBETA*F/A,
     2  F9)
      TS1=BETAZ*F/(WN*W2)
      TS2=F/(WN*W2)
      PRINT 48,NALPH,TS1,ALPHA ,TS2
48    FORMAT(15X*NALPH *I2,17X*JUDGED TO BE:*4X,13HBETA*F/(K*A2),F9.5/
     1 15X*ALPHA*F11.8*/CM*1X*CONTOURS ++ + - -=* 9X,8HF/(K*A2),F9.5)
      TS1=ALPHA*WN*W2
      PRINT 50,NBETA,TS1,BETAZ
50    FORMAT(15X*NBETA*I3,12X*L AND V  ++ + - -=*7X,10HALPHA*K*A2,F9.5/
     1  15X*BETA*F9.2//)
      PRINT 52
52    FORMAT(15X*CONTOUR*6X*I*9X*A*6X*I(VAC)*3X*A(VAC)*3X*I(REL)*3X
     1  *A(REL)*)
      SUMRAT=0
      DO 56  I=2,8,3
      J=10-I
      AVGINT=SUMA(J)/AREA(J)
      VACINT=POWER*(1.-CONMIN(J))/(PI*W2*LOGF(1./CONMIN(J))*D)*EXPF(*ALP
     1   HA*WN*W2*ZETA)
      VAREA=PI*W2*LOGF(1./CONMIN(J))*D
      RATIOI=AVGINT/VACINT
      SUMRAT=SUMRAT+RATIOI
      RATIOA=AREA(J)/VAREA
      PRINT 54,CONMIN(J),AVGINT,AREA(J),VACINT,VAREA,RATIOI,RATIOA
54    FORMAT(15X,F5.1,F12.3,F10.2,F9.2,F9.2,F9.2,F8.3,F9.3)
56    CONTINUE
      AVGRAT=SUMRAT/3.
      TS1=QQQMAX/1000.
      TS2=(POWER/1000.)*EXPF(*ALPHA*WN*W2*ZETA)/(PI*(F/(WN*WIDTH))**2)
      TS3=TS1/TS2
      TS4= -10.*ALOG10(TS3)
      TS5= -10.*ALOG10(AVGRAT)
      PRINT 58,AVGRAT,TS1,TS2,TS3,TS4,TS5
58    FORMAT(15X,45X,F8.3//15X*WITH BLOOMING ICP(F) =*F8.3* KW/CM2*/15X
     1  *WHILE IN VACUUM IT IS*F9.3 * KW/CM2: I(REL) =*F6.3/
     2  15X,28X*DBLOSS =*F7.3* TO*F7.3/)
      TS1=RUNPP/1000.
      TS2=(RUNPP/QQGMAX-1.)*100.
      TS3=(1.-ZPP/FL)*100.
      TS4=DEF*WIDTH*FOCUS
      TS5=(TS4/(Z*100000.))*1000.
      PRINT 60,TS1,TS2,ZPP,TS3,TS4,TS5
60    FORMAT(15X,13X*I PEAK POWER =*F7.3* KW/CM2 (*F4.1* 0/0 ABOVE ICP(F
     1))*/15X,12X*OCCURS AT ZPP =*F7.3* KM*5X*(*F4.1* 0/0 BEFORE F)*/
     2  15X,15X*DEFLECTION =*F7.3* CM*5X*(*F6.3* MRAD)*//15X*COMMENTS:*
     3  /////)
      IF(NDIF .EQ. 0) 61,62
61    PRINT 615
615   FORMAT(62X*IMPLICIT, CDA*)
```

UNCLASSIFIED

```
      GO TO 63
62    PRINT 625
625   FORMAT(62X*IMPLICIT, SPLINES*)
63    JSEQ=MPRISEQ(1)
      CALL DATE(MONTH,IDAY,IYEAR,JULDAY)
      PRINT 64,JSEQ,MONTH,IDAY,IYEAR
64    FORMAT(15X,47X*SEQUENCE *04/15X,47X*DATE*5X,I2*/*
     1  I2*/*I2/15X,47X*RUN TIME     !*////)
66    CONTINUE

      RETURN
      END
```