NRL Report 7681

Documentation of PROP E, a Computer Program for the Propagation of High-Power Laser Beams Through Absorbing Media

PETER B. ULRICH, JOHN N. HAYES, J. HARRISON HANCOCK, AND JEANNE T. ULRICH

Optical Radiation Branch
Optical Sciences Division

May 29, 1974

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER NRL Report 7681	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DOCUMENTATION OF PROP E, A COMPUTER PROGRAM FOR THE PROPAGATION OF HIGH-POWER LASER BEAMS THROUGH ABSORBING MEDIA		5. TYPE OF REPORT & PERIOD COVERED Final report on one phase of a continuing NRL problem. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Peter B. Ulrich, John N. Hayes, J. Harrison Hancock, and Jeanne T. 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 54
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 propagation
Nonlinear optics
Atmospheric optics
Computer simulation

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

This report details the numerical methods employed in a solution of the problem of high-power laser-beam propagation. A transformation of the partial differential equation for the light beam is used which differs from that of a previous study of the problem ("Propagation of High-Energy 10.6 Micron Laser Beams Through the Atmosphere," A. H. Aitken, J. N. Hayes, and P. B. Ulrich, NRL Report 7293, May 28, 1971) and which offers a versatility allowing a study of highly distorted beams to be made. A detailed description of the subroutines and their operation is given along with flow charts and FORTRAN listings.

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)

CONTENTS

DOCUMENTATION OF PROP E, A COMPUTER PROGRAM FOR THE PROPAGATION OF HIGH-POWER LASER BEAMS THROUGH ABSORBING MEDIA

INTRODUCTION

In a previous report (1), here after referred to as AHU, a description was presented of the physics of high-power laser-beam propagation through the atmosphere, together with a discussion of a numerical method for integrating the appropriate equations. A detailed listing of the computer code itself was not presented in that report. It is the purpose of this report to present the numerical methods in considerable detail. However, since the development of the initial computer program, revisions have been made in the form of the basic differential equation to be solved and in the numerical methods used; it is this latest version which is described here. A numerical method wholly different from the present one, but much like that of Bradley and Herrmann (2), has also been developed and will be described in a separate report.

Because the physics of the problem solved by the present code is the same as that discussed in AHU, the discussion here will concentrate on the changes that have been introduced and on the structure of the program. Nevertheless, the physics of the problem will, of necessity, be present throughout the discussion and will be assumed to be understood. Also the limitation of numerical methods for solving problems of nonlinear wave propagation will be discussed.

FORMULATION OF THE PROBLEM

The basic equation to be solved is a partial integrodifferential nonlinear equation, whose derivation was discussed in AHU, one form of which is

$$2ik\frac{\partial\varphi}{\partial z} + \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = \frac{3N(\gamma-1)k^2\alpha P}{c_s^2 v_0}\,\frac{e^{-\alpha z}}{1+\frac{\Omega z}{v_0}}$$

$$\times\ \varphi\int_{-\infty}^{0} ds\,(1-\delta)\exp\left[\frac{s}{v_0 t\left(1+\frac{\Omega z}{v_0}\right)}\right]|\varphi(\vec{r}+s\hat{v}_0)|^2, \qquad (1)$$

where

$$\iint_{z=\mathrm{const}} dxdy\ \varphi^*(x,y,z)\varphi(x,y,z) = 1, \qquad (2)$$

Note: Manuscript submitted October 26, 1973.

and the intensity at the point whose coordinates are (x, y, z) is

$$I(x, y, z) = Pe^{-\alpha z}\varphi^*(x, y, z)\varphi(x, y, z). \tag{3}$$

In Eqs. (1) through (3), z is taken as the direction of beam propagation, and the derivation of Eq. (1) from the wave equation involves the Fresnel approximation, among other assumptions about the physics.

Because a focussed laser beam changes its transverse linear dimensions drastically between the aperture and the focal point, it is necessary, in numerical methods where the amplitude can be computed only at a finite number of points, to have three-dimensional grids whose transverse dimensions shrink in size as the focal point is approached. For a vacuum beam the actual beam size can be determined either analytically or by numerically integrating the Fresnel integral. For the nonlinear problem the actual desired shrinkage of the transverse mesh is determined by the solution to the problem itself. Herein lies the source of a difficulty in calculating a numerically reliable solution. A second difficulty is that the beam amplitude has two parts which must be determined, the real and the imaginary parts, or equivalently, the modulus and the phase. Because the intensity, and therefore the modulus, of the wave amplitude evolves slowly with z and is a "relatively" smoothly developing function in its transverse dimensions, this portion of a calculation would proceed without complication were it not that the determination of the modulus is closely intertwined with that of the phase. If the geometric surfaces to which the grids used in the numerical calculations are attached do not closely match surfaces of constant phase, then even though the grid shrinks in accordance with the actual beam size, the real and imaginary parts will soon develop high-frequency components which will ultimately exceed the Nyquist frequency, and incorrect results are calculated thereafter. It is therefore highly desirable to have a three-dimensional grid in space that simultaneously changes its shape in accordance with the actual beam size and adapts to the surfaces of constant phase. Since both requirements need the actual solution to the differential equation, Eq. (1), which is the purpose of the calculation, there are two general ways one can proceed. One is to solve, on a step-by-step basis, for the solution and the grid system itself; a second is to make phase transformations of the solution to Eq. (1) before the numerical work has begun that automatically shrink the coordinate system and remove to the first order the troublesome phase oscillations in the real and imaginary parts of the amplitude. Experience has shown that whenever beam distortions are so severe that the second method fails, the solutions are not of sufficient practical interest to go to the first approach to the problem. Therefore, attention has been confined to the second approach.

In AHU the grids were placed on concentric spherical surfaces whose centers lay on the axis of propagation a distance l from the aperture. The concomitant changes in the differential equation solved the dimensional and phase problems for a wide variety of physical situations; in this form the distance l was a free parameter that was adjustable to give the best solution consistent with the numerical criteria discussed in that report. The numerical calculations were compared with laboratory experiments (3) and found to be in excellent agreement, providing a high confidence level in the theoretical predictions. Nevertheless the grid scheme used in AHU had sufficient limitations that another scheme was sought to supplement the original computer program. One of the problems of the original program can be seen easily by setting the free parameter l equal to the focal point; then all the grid points at $z = l = f$ collapse to a point, which is correct for geometric optical beams but is not correct for wave beams. To avoid this singular behavior in the coordinate system, the surfaces of constant phase, which are concentric spheres, will be replaced by the surfaces of constant phase of a vacuum Gaussian wave beam focussed at f or by surfaces of constant phase very close to these. Simultaneous with this transformation will be a transformation to transverse variables which change size in accordance with a vacuum Gaussian wave beam focussed at f or with a beam very similar in size to this beam. This description of the desired transformations will be presented mathematically in the following.

The wave amplitude of a scalar Gaussian wave field is given by

$$\varphi_{GV}(x, y, z) = \frac{-i\left[\frac{z}{ka_0^2} + i\left(1 - \frac{z}{f}\right)\right]}{\sqrt{\pi a_0^2}\, D_V(z)} \exp\left\{-\frac{x^2 + y^2}{2a_0^2 D_V(z)} + \frac{ik(x^2 + y^2)}{2z}\left(1 - \frac{1 - \frac{z}{f}}{D_V(z)}\right)\right\}, \tag{4}$$

where

$$D_V(z) = \frac{z^2}{k^2 a_0^4} + \left(1 - \frac{z}{f}\right)^2 . \tag{5}$$

The function φ_{GV}, given by Eq. (4), solves Eq. (1) with the nonlinear term set equal to zero. Here a_0 represents the radius from the center of the beam at the aperture at which the intensity drops by a factor of e from its peak value in the center of the aperture, f is the focal distance from the aperture, and k is the wave number. The transverse linear dimension $a_0\sqrt{D_V(Z)}$ is the e^{-1} – power point at a distance z from the aperture. The beam retains its Gaussian shape for all values of z, since

$$I(x, y, z) = |\varphi_{GV}(x, y, z)|^2 = \frac{\exp\left[-\frac{x^2 + y^2}{a_0^2 D_V(z)}\right]}{\pi a_0^2 D_V(z)} . \tag{6}$$

The peak intensity in such a beam occurs at a range $Z_w(V)$ given by

$$Z_w(V) = \frac{f}{1 + \frac{f^2}{k^2 a_0^4}}, \tag{7}$$

whereas the e^{-1} – power point at $Z_w(V)$ is given by

$$a_w(V) = \frac{\frac{f}{ka_0}}{\sqrt{1 + \frac{f^2}{k^2 a_0^4}}} . \tag{8}$$

The function $D_V(Z)$ can be written as

$$D_V(z) = \frac{a_w^2(V)}{a_0^2}\left\{1 + \left[\frac{a_0^2}{a_w^2(V)} - 1\right]\left[1 - \frac{z}{Z_w(V)}\right]^2\right\}; \tag{9}$$

$Z_w(V)$ will be called the location of the "waist" of the vacuum beam, and $a_w(V)$ will be termed the radius at the beam waist.

It is well known that an initially Gaussian beam that is launched from the aperture will not remain so when thermal blooming occurs, i.e., when the nonlinear term in Eq. (1) is retained and the initial conditions are given by Eq. (4) with Z set equal to zero. Two gross features that crudely describe the thermally bloomed beam are the waist location Z_w and some measure a_w of the beam size at the waist. The waist of the thermally bloomed beam occurs at a shorter distance ($Z_w \leqslant Z_w(V)$) than does that for the vacuum beam, and the beam size is generally larger, i.e., $a_w \geqslant a_w(V)$. Therefore, $D_V(Z)$ is accordingly generalized to

$$D(z) = \frac{a_w^2}{a_0^2}\left[1 + \left(\frac{a_0^2}{a_w^2} - 1\right)\left(1 - \frac{z}{Z_w}\right)^2\right]. \tag{10}$$

The parameters a_w and Z_w are a generalization of the parameter l chosen in AHU. That limiting case is achieved by taking $a_w \to 0$ and $Z_w = Z_w(V)$.

To facilitate the discussion of the transformation on the dependent variables, one can set

$$S(\mathbf{r}) = -\frac{3N(\gamma - 1)}{C_s^2}\,\frac{k^2\alpha P}{v_0}\,\frac{e^{-\alpha z}}{1 + \frac{\Omega z}{v_0}}\int_{-\infty}^{0} ds\,|\varphi(\mathbf{r} + S\mathbf{v}_0)|^2 \tag{11}$$

and rewrite Eq. (1) as

$$2ik\,\frac{\partial\varphi}{\partial z} + \nabla_\perp^2\varphi + S(\mathbf{r})\varphi(\mathbf{r}) = 0\,. \tag{12}$$

The following transformation to a new dependent variable $\varphi_1(x, y, z)$ is equivalent to referring the beam to surfaces of constant phase approximating the vacuum Gaussian beam:

$$\varphi(x, y, z) = \frac{\varphi_1(x, y, z)\left\{\exp\left[-\frac{k(x^2 + y^2)}{2}\,\frac{d\ln\sqrt{D(z)}}{dz}\right]\right\}}{\sqrt{a_0^2 D(z)}}\,. \tag{13}$$

From Eqs. (2) and (13) it follows that

$$\iint \frac{dxdy}{a_0^2 D(z)} \; |\varphi_1(x, y, z)|^2 = 1 , \tag{14}$$

and the equation for $\varphi_1(x, y, z)$ is

$$2ik \frac{\partial \varphi_1}{\partial z} + \nabla_\perp^2 \varphi_1 + 2ik \frac{d \ln \sqrt{D(z)}}{dz} \left(x \frac{\partial \varphi_1}{\partial x} + y \frac{\partial \varphi_1}{\partial y} \right)$$

$$+ \left\{ S(\mathbf{r}) - k^2(x^2 + y^2) \left[\frac{d^2 \ln \sqrt{D(z)}}{dz^2} + \left(\frac{d \ln \sqrt{D(z)}}{dz} \right)^2 \right] \right\} \varphi_1(x, y, z) = 0 ,$$

but

$$\frac{d^2 \ln \sqrt{D(z)}}{dz^2} + \left(\frac{d \ln \sqrt{D(z)}}{dz} \right)^2 = \frac{1}{D^2} \frac{a_w^4}{Z_w^2 a_0^2} \left(\frac{a_0^2}{a_w^2} - 1 \right) , \tag{15}$$

so that the equation for $\varphi_1(x, y, z)$ becomes

$$2ik \frac{\partial \varphi_1}{\partial z} + \nabla_\perp^2 \varphi_1 + 2ik \frac{d \ln \sqrt{D}}{dz} \, \mathbf{r}_1 \cdot \nabla_1 \varphi_1$$

$$+ \left[S(\mathbf{r}) - \frac{x^2 + y^2}{a_0^4 D^2} \frac{k^2 a_w^4}{z^2} \left(\frac{a_0^2}{a_w^2} - 1 \right) \right] \varphi_1(x, y, z) = 0 .$$

In terms of φ_1, S becomes

$$S(\mathbf{r}) = -\beta \frac{e^{-\alpha z}}{1 + \frac{\Omega z}{v_0}} \frac{1}{a_0^3 D(z)} \int_{-\infty}^{0} ds \; |\varphi_1(\mathbf{r})|^2 , \tag{16}$$

where

$$\beta = \frac{3N(\gamma - 1)}{C_s^2} \frac{k^2 \alpha P a_0}{v_0} . \tag{17}$$

Next, an independent coordinate transformation is introduced that enables the numerical grid system to shrink in approximate accordance with the beam:

$$\tilde{x} = \frac{x}{a_0 \sqrt{D(z)}} , \tag{18a}$$

$$\tilde{y} = \frac{y}{a_0 \sqrt{D(z)}}, \qquad (18b)$$

and

$$\varphi_2(\tilde{x}, \tilde{y}, z) \equiv \varphi_1(x, y, z). \qquad (18c)$$

The differential equation for φ_2 becomes

$$2ik \frac{\partial \varphi_2}{\partial z} + \tilde{\nabla}_\perp^2 \varphi_2 + \left[S(x(\tilde{x}), y(\tilde{y}), z) - \frac{k^2 a_w^{\,4}}{Z_w^{\,2}} \left(\frac{a_0^2}{a_w^{\,2}} - 1 \right) \frac{\tilde{x}^2 + \tilde{y}^2}{a_0^2 D} \right] \varphi_2(\tilde{x}, \tilde{y}, z) = 0 ; \qquad (19)$$

thus

$$\iint d\tilde{x} d\tilde{y} \, |\varphi_2(\tilde{x}, \tilde{y}, z)|^2 = 1, \qquad (20)$$

and

$$S = -\beta \frac{e^{-\alpha z}}{1 + \frac{\Omega z}{v}} \frac{1}{a_0^2 \sqrt{D}} \int_{-\infty}^{0} d\tilde{s} \, |\varphi_2(\tilde{x}, \tilde{y}, z)|^2 . \qquad (21)$$

Finally, one last transformation is made, this time on the independent variable Z:

$$\tilde{Z}(z) = \frac{1}{k a_0^2} \int_0^z \frac{dz'}{\sqrt{D(z')}} . \qquad (22)$$

The integration in Eq. (22) can be done from tables; the result is

$$\tilde{Z} = \frac{Z_w}{k a_0 a_w} \frac{1}{\sqrt{\frac{a_0^2}{a_w^2} - 1}} \left[\sinh^{-1} \sqrt{\frac{a_0^2}{a_w^2} - 1} - \sinh^{-1} \sqrt{\frac{a_0^2}{a_w^2} - 1} \left(1 - \frac{z}{Z_w} \right) \right] ; \qquad (23)$$

the inverse of Eq. (23) is

$$z = Z_w \left[1 + \frac{1}{\sqrt{\frac{a_0^2}{a_w^2} - 1}} \sinh \left(\frac{k a_0 a_w}{Z_w} \sqrt{\frac{a_0^2}{a_w^2} - 1} \, \tilde{Z} - \sinh^{-1} \sqrt{\frac{a_0^2}{a_w^2} - 1} \right) \right] . \qquad (24)$$

By defining

$$\psi(\tilde{x}, \tilde{y}, \tilde{Z}) \equiv \varphi_2(\tilde{x}, \tilde{y}, z(\tilde{Z}))$$

and

$$S(\tilde{x}, \tilde{y}, \tilde{Z}) \equiv S(x(\tilde{x}), y(\tilde{y}), z(\tilde{Z})), \tag{25}$$

the differential equation for ψ becomes

$$2i \frac{\partial \psi}{\partial \tilde{Z}} + \frac{1}{\sqrt{D(Z(\tilde{Z}))}} \tilde{\nabla}_{\perp}^2 \psi + \left[\tilde{S} - \frac{k^2 a_w^4}{Z_w^2}\left(\frac{a_0^2}{a_w^2} - 1\right) \frac{\tilde{x}^2 + \tilde{y}^2}{\sqrt{D(z(\tilde{Z}))}}\right] \psi = 0, \tag{26}$$

and

$$\iint\limits_{\tilde{Z}=\text{const}} d\tilde{x} d\tilde{y} \, |\psi(\tilde{x}, \tilde{y}, \tilde{Z})|^2 = 1. \tag{27}$$

Equation (26) is the actual one solved in the computer program described in this report. The initial conditions on ψ remain to be specified. If at $z = 0$, i.e., at the aperture, the initial amplitude $\psi(x, y, z)$ is given by

$$\varphi(x, y, 0) = f(x, y), \tag{28}$$

then

$$\psi(\tilde{x}, \tilde{y}, 0) = a_0 \exp\left[\frac{ika_0^2(\tilde{x}^2 + \tilde{y}^2)}{2Z_w}\left(1 - \frac{a_w^2}{a_0^2}\right)\right] f(a_0\tilde{x}, a_0\tilde{y}). \tag{29}$$

The transformations used here are similar to, but not identical with, those used by C. B. Hogge* and by J. Hermann and L. C. Bradley (2).

The stability condition for numerical integration of the equation of propagation Eq. (26), according to the Harmuth technique (see AHU, for example) now becomes

$$\Delta\tilde{Z} \leqslant \left\{\frac{4}{\sqrt{D}}\left[\frac{1}{(\Delta\tilde{x})^2} + \frac{k^2 a_w^4}{Z_w^2}\left(\frac{a_0^2}{a_w^2} - 1\right) n^2(\Delta\tilde{x})^2 + \frac{\beta e^{-\alpha z}}{1 + \frac{\Omega z}{v_0}}\right]\right\}^{-1}. \tag{30}$$

If $\Omega/v_0 > 0$,

*Private communication.

$$\frac{Z_w}{Z_w(V)} \equiv P \tag{31a}$$

and

$$\frac{a_w}{a_w(V)} \equiv Q. \tag{31b}$$

Then with $\hat{f} = f/ka_0^2$,

$$\Delta\widetilde{Z} \leqslant \left\{\beta + \frac{4}{Q\hat{f}}\left[\frac{1}{(\Delta x)^2} + Q^2P^2n^2(\Delta x)^2\right]\right\}^{-1}. \tag{32}$$

Here n is the number of mesh points along the axes. The number of steps in the direction of propagation required to reach the focal point is called NSTEP and is given by

$$\text{NSTEP} = \frac{\widetilde{Z}(f)}{\Delta\widetilde{Z}}. \tag{33}$$

When $P = Q = 1$, the coordinate system matches with the vacuum Gaussian beam. For a wide variety of cases, experience has shown that, in using the code, P generally should be taken to be unity or very near unity but that Q may deviate rather drastically from unity.

Both in this coordinate system, in that described in AHU, and indeed in all systems that involve focussing coordinate systems, the physics of the problem adds a computational problem not yet addressed. When thermal blooming occurs, the beam will deviate from the initial direction of propagation and head into the wind. Therefore, the "real beam" may not "choose" to stay within the confines of the finite grid set up in three-dimensional space in accordance with the previous discussion. The consequences of this phenomenon were discussed in AHU. To avoid the certain numerical catastrophes connected with this problem and because the prior transformation does apply to the whole of space, the computer program includes a subroutine FOLLOW which senses the location of the peak-intensity point in the beam and lets the transverse grid center itself on this point. In this manner the beam is kept from "hitting the edges of the grid mesh" in the numerical computations.

The transformations discussed in the previous paragraph and Eq. (26) constitute the basic difference between this version of the code and the earlier one. Details of the code operation and a code listing are provided in the next two sections. However, it is not to be expected that the new formulation will adequately treat all problems that arise. Restrictions on the values of derivatives at the aperture are still in force, as reported in AHU. In addition, for those cases which involve severe blooming, the distortions may be so extreme as to place contradictory requirements on the mesh size in the two orthogonal directions, or a highly refined mesh may be required which will greatly increase running time on the computer or exceed the memory capacity of the computer or both. Experience has indicated that these problems are somewhat academic, however, in that the conditions that optimize the peak intensity in the focal plane, which is frequently the object of interest, occur at relatively mild beam distortions.

DESCRIPTION OF COMPUTER PROGRAM PROP E

Introduction

The purpose of the computer program is to solve Eq. (26) numerically. The desired output from the program described here comprises the real and imaginary parts of the amplitude at select ranges from the aperture. From these data are computed the intensity distribution at these same ranges, the location of the point (or points) of maximum intensity, the deflection of this same point into the wind, contour plots showing isophotes and printouts that give average intensities and powers inside prescribed isophotes. For numerical purposes the right-hand side of Eq. (27) is evaluated as one check on the computation. (For a description of checks on numerical accuracy, see AHU.) These data at select ranges should be stored in arrays to plot, in a given run, deflection vs range, maximum intensity vs range, and relative intensities (i.e., computed maximum intensity divided by $I_{VAC}(0, 0, Z)e^{-\alpha Z}$ and like ratios for quantities averaged inside isophotes) vs range. In the actual execution of the program, virtually all of these quantities must be calculated at a number of ranges much larger than the number of ranges for which the user wants these data; therefore, flags and counters must be set up to indicate the frequency with which printouts and plots should be made. Because of the variety of ways the input data can be manipulated, options for input-data methods are included. How this is done will be described in this section; the actual code listing will be deferred to the next section. A flow chart is provided as Appendix A.

Program statements that are no longer used may confuse the reader; these will be pointed out wherever they occur. Since their existence affects neither the operation nor the running time of the code, they have not been systematically extirpated.

PROP E is organized into a main program and sixteen subroutines; the listings are provided as Appendix B. The letter E stands for the fact that Eq. (26) is solved by an explicit algorithm, a two-point predictor; PROP stands for propagation. The main part of the program defines the variables and common arrays and calls on the subroutines to do the actual calculations.

Main Program

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. Since the comment cards in the listing provide terse definitions of the quantities involved, these will be amplified in some measure here.

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 specifies the array. These arrays are:

CC: Index of refraction changes, meaning the value of the entire coefficient of ψ in the nonlinear term in Eq. (26). This array is calculated in subroutine INDEX.

PLTGSN: The PLTGSN array calculates the parameter N of Gebhardt and Smith (4); it is a relic from efforts to compare the prediction of the computer program with the experimental results of Gebhardt and Smith and has not been deleted subsequently from the code.

PLTRAT: The PLTRAT 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 $a^{-\alpha Z}$; μ is the fraction of the peak irradiance for which an isophate is calculated; μ here is taken to be 0.2, 0.5, and 0.8. These values are plotted at the conclusion of the calculation in subroutine GRAPH 2 but printed out in tabular form at select ranges in subroutine OUTPUT.

PLTREL: The relative intensity at Z is defined as the peak intensity in the beam at range Z divided by the peak intensity of the vacuum beam at Z times $e^{-\alpha Z}$. These are printed in subroutine OUTPUT and plotted in subroutine GRAPH 2.

NOTE OF WARNING: In the listing of the present code, subroutine VACAMP defines the vacuum beam, here taken to be a Gaussian beam. For this case, $I_{\text{peak}}(V) = I_V(0, 0, Z)$; for beams such that at Z the on-axis intensity vanishes, $I_{\text{Rel}} \to \infty$, and a diagnostic will occur.

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

COMMON PARAMETERS

D: Holds the values of

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

DNEW: Holds the values of

$$D(z) = \frac{a_0^2}{a_w^2} \left[1 + \left(\frac{a_w^2}{a_0^2} - 1\right)\left(1 - \frac{z}{z_w}\right)^2\right].$$

FHAT: $\hat{f}$ = FHAT = f/ka_0^2.

FOCUS: $\sqrt{\text{DNEW}}$.

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

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

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

IMAX and JMAX: The indices I and J label the points in the arrays transverse to the beam axis, where $I = 1, 2, \ldots, NX + 1$ and $J = 1, 2, \ldots, NY + 1$; (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 $|\iint d\tilde{x}d\tilde{y}\,|\psi|^2 - 1| > \text{ECHNG}$; ECHNG is calculated in subroutine INPUT.

If NOUT $\neq$ 1, this option is not exercised, and the program will continue to compute even though Eq. (27) ceases to be satisfied.

NSTEP: This is defined in the section "FORMULATION OF THE PROBLEM". The program automatically computes NSTEP as well as HZ in subroutine SETUP.

SUM: 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. The main program functions as an executive routine, the detailed computations being relegated to the subroutines. Iteration in the Z variable is also done within the main program as are tests on whether or not Eq. (27) is satisfied to within prescribed tolerances.

Subroutines

INPUT

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

HX, HY correspond to $\Delta\tilde{X}$ and $\Delta\tilde{Y}$. For Gaussian beams, HX, HY $\approx$ 0.18 have been found to be adequate for distortions that are not too severe; in the latter case, more mesh points are needed, i.e., larger NX and NY together with smaller values of HX and HY.

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

NOLPLC is a remnant that involved code testing (see AHU). Always set NOLPLC = 0.

LPLOT is a flag to provide either group plotting or sequential plotting. 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 performed, this is a space-saving device. Sequential plotting means that the contour plots (and others) are each plotted separately.

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

ZT define range values for which the absorption coefficient is set equal to zero. This was useful for comparisons with experiment and for theoretical study.

DIAM is defined to be $2\sqrt{2a_0}$ and 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 is the elevation angle for beams which are launched not parallel to the surface and is read in degrees.

WN is the wavenumber $k = 2\pi/\lambda$.

Since the determination of the absorption coefficient through the specification of P, PH20, and T by the code (see AHU; also see subroutine SETUP) is not always convenient or appropriate, a set of options are allowed. These options 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 through 4. Finally, INPUT Prints the input data.

SETUP

Refer to Appendix B for a description of this subroutine.

INITL

Subroutine INITL calculates the real and imaginary parts of the beam amplitude at the aperture ($\widetilde{Z} = 0$) for each point in the grid and stores the values in the appropriate array. Using the Eulerian method, it does the same at $\widetilde{Z} = \Delta\widetilde{Z} = H_z$. This subroutine needs the initial beam data provided by subroutine VACAMP.

VACAMP

Subroutine VACAMP computes the vacuum beam everywhere; in particular it is called on by subroutine INITL to set up the boundary conditions for the problem. In addition, though subroutine INTENS does not explicitly call on 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, 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 must be provided for in a table. In the latter case the CONMIN table should be removed, since it becomes meaningless, or should be corrected if the vacuum beam at nonzero values of Z is to be determined analytically.

PREDCT

See Appendix B for the comment cards and AHU for a discussion of the predictor.

UPDATZ

See Appendix B for the comment cards.

BOUNDS

Assigns 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 zero to the boundary values.

SHIFT

See Appendix B for the comment cards.

FOLLOW

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

WARNING: This subroutine is incorrect when an array has an explicit dependence on the variable Y; this case occurs when kinetic cooling is present, i.e., $\delta \neq 0$ and $\tau \neq 0$. However, it has been determined experimentally (5) that τ is substantially smaller (6) than it had been thought originally to be (7). Therefore, for CO_2 laser beams, δ has been consistently set equal to zero; for beams other than CO_2, the kinetic-cooling theory is inappropriate anyway, so again δ is set equal to zero. Under these conditions the subroutine correctly shifts the arrays.

INTENS

See Appendix B for the comment cards. Subroutine INTENS also prints the array that compares intensities averaged over isoirradiance contours with the corresponding vacuum beam quantities, the areas of these regions, etc. (See the previous comments in subroutine VACAMP.)

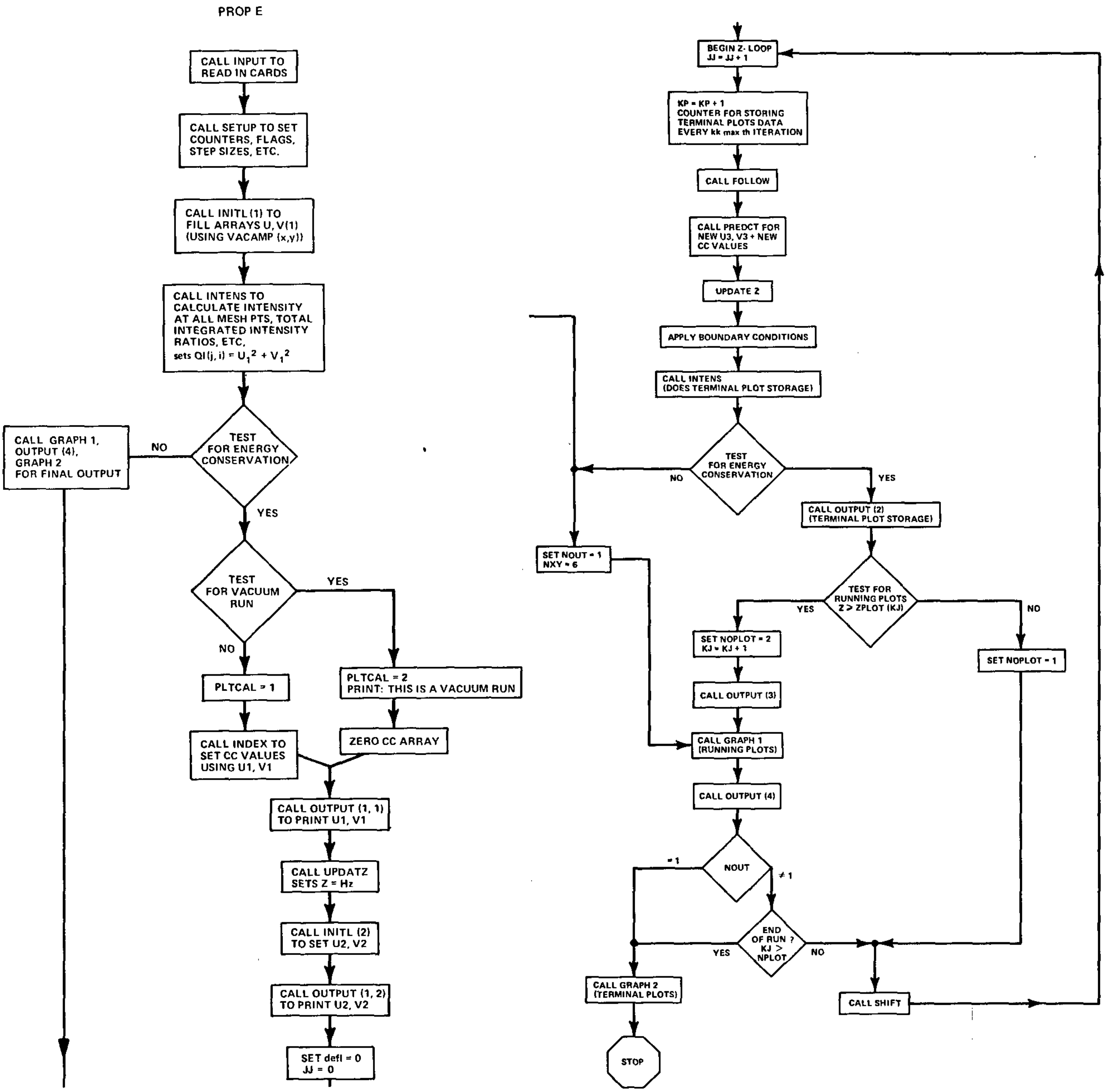
The remaining subroutines provide graphical output and are machine dependent. Therefore, these would have to be replaced by equivalent subroutines appropriate for use on the user's computer. This program has been written to be used on the CDC 3800 at the Naval Research Laboratory.

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.
2. J. Herrmann and L.C. Bradley, Lincoln Laboratory Report LTP-10, July 12, 1971.
3. P.B. Ulrich, J.N. Hayes, and A.H. Aitken, *J. Opt. Soc. Amer.* 62, 298 (1972).
4. F.G. Gebhardt and D.C. Smith, *IEEE J. Quantum Electronics* QE-7, 63 (1971).
5. L. Sica, *Appl. Phys. Lett.* 22, 396 (1973).
6. H.E. Bass and H. Bauer, J. Appl. Optics, in preparation.
7. A.D. Wood, M. Camac, and E.T. Gerry, *J. Appl. Opt.* 10, 1877 (1971).

Appendix A

FLOW CHART



Appendix B

FORTRAN LISTINGS

```
      PROGRAM PROP E

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 ALTER
C      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         THE ALGORITHM USED TO INTEGRATE IN Z IS A SIMPLE TWO-POINT
C      PREDICTOR.  SYMMETRIC SECOND-ORDER DIFFERENCES REPRESENT THE
C      TRANSVERSE LAPLACIAN; THE INTEGRAL ALONG THE WIND TO CALCULATE
C      THE CHANGE IN INDEX OF REFRACTION IS DONE BY THE TRAPEZOIDAL RULE.
C      SINCE THE ALGORITHM IN Z IS SECOND ORDER, SPECIFICATION OF THE
C       AMPLITUDE AT THE FIRST MESH PLANE AS WELL AS AT Z=0 IS REQUIRED.

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

C     COMMON ARRAYS

C         CC          INDEX OF REFRACTION CHANGES
C         FORM1       PRINTING STATEMENT FORMATS
C         FORM2       PRINTING STATEMENT FORMATS
C         PLTGSN      GEBHARDT AND SMITH N-PARAMETER VALUES FOR PLOTTING
C         PLTRAT      RATIOS OF INTENSITY TO VACUUM INTENSITY FOR PLOTTING
C         PLTREL      RELATIVE INTENSITY VALUES FOR PLOTTING
C         QI          INTENSITY IN A TRANSVERSE PLANE AS A FUNCTION OF
C                     X AND Y
C         QJMAX       BEAM DEFLECTIONS
C         QQMAX       MAXIMUM INTENSITY VALUES
C         U,V         REAL,IMAGINARY PARTS OF THE AMPLITUDE
C         ZI          POINTS AT WHICH INTENSITY RATIOS ARE CALCULATED
C         ZM          POINTS AT WHICH INTENSITY MAXIMA AND DEFLECTIONS
C                     ARE CALCULATED ........IN KILOMETERS
C         ZPLOT       VALUES OF Z (KMS) AT WHICH PLOTS ARE TO BE DONE

C     COMMON VARIABLES AND CONSTANTS GENERATED BY THE PROGRAM
```

```
C       ALFCON = STORES INITIAL VALUE OF ALPHA
C       ALFSUM = LOCATION FOR ALPHAINTEGRATION SUMMING DURING Z-ITERATIONS
C       AWAIST = RADIUS OF BEAM AT WAIST IN VACUUM
C       CZERO  = STORES INITIAL VALUE OF C
C       D      = VACUUM RAY TRAJECTORY (DISTANCE FROM CENTER) SQUARED
C       DEF    = DEFLECTION OF BEAM IN PROGRAM UNITS, MEASURED ALONG Y-DIRECTIO
C                (BUT Y-DIRECTION IS PLOTTED ON X-AXIS)
C       DEFL   = DEFLECTION OF BEAM IN CM   = DEF*WIDTH*FOCUS
C       DNEW   = D CALCULATED IN TERMS OF AWAIST AND ZWAIST
C       FACTOR = SQUARE ROOT OF D
C       FHAT   = ZETA EVALUATED AT THE FOCAL POINT
C       FL     = FOCAL LENGTH IN KMS
C       FOCUS  = FACTOR FOR SHRINKING COORDINATE SYSTEM
C       HZ     = STEP SIZE IN THE Z DIRECTION (IN PROGRAM UNITS)
C       HZX    = HZ/HX**2
C       HZY    = HZ/HY**2
C       IMAX AND JMAX = INDICES OF LARGEST QI(I,J), (PEAK INTENSITY)
C       IQ     = INDEX FOR INTENSITY MAXIMUM (QQMAX)
C       JCMAX  = J-INDEX OF LARGEST QI(NX,J) FOR GIVEN NX
C       JMAX     (SEE IMAX)
C       KJ     = ZPLOT INDEX
C       KKMAX  = NUMBER OF STEPS BETWEEN TERMINAL PLOT CALCULATIONS
C       KP     = COUNTER FOR TERMINAL PLOT CALCULATIONS
C       NOPLOT = PLOTTING FLAG -- =1 TO SKIP PLOT, =2 TO PLOT
C       NOUT   = ENERGY ABORT FLAG
C       NSTEP  = NUMBER OF STEPS IN THE Z-DIRECTION TO LAST ZPLOT VALUE
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       QMAX   = VALUE OF PEAK INTENSITY, QI(JMAX,IMAX)
C       ROZERO = STORES INITIAL VALUE OF RO
C       SUM    = TOTAL INTEGRATED INTENSITY
C       SUM1   = SAVES INITIAL SUM FOR SUM2 CALULATION
C       SUM2   = TEST FOR ENERGY CONSERVATION
C       WIDTH  = WIDTH IN CM  OF A PROGRAM UNIT IN THE X-Y PLANE (INITIAL RADIUS
C       W2     = INITIAL RADIUS SQUARED
C       Z      = DISTANCE ALONG BEAM PATH IN KILOMETERS
C       ZETA   = RANGE OVER 1/2 RAYLEIGH RANGE = Z/(WN*W2)
C       ZWAIST = RANGE TO WAIST OF BEAM IN VACUUM
C       ZZ     = DISTANCE ALONG BEAM PATH  (IN PROGRAM UNITS)


      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ
```

```
      CALL INPUT
      CALL SETUP
      CALL INITL (1)
      CALL INTENS

C     TEST FOR ENERGY CONSERVATION
      IF (ABSF((SUM-POWER)/ROWER)•GT•ECHNG) GO TO 5

C     TEST FOR VACUUN RUN AND FILL INDEX OF REFRACTION ARRAY, CC
      IF (BETA.EQ.0) GO TO 1
      PLTCAL=1
      CALL INDEX (1)
      GO TO 4

C     ZERO INDEX OF REFRACTION ARRAY, CC, FOR VACUUM
    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

C     FINISH CALCULATING AND PRINTOUT INITIAL CONDITIONS
    4 CALL OUTPUT (1,0,1)
      CALL UPDATZ
      CALL INITL (2)
      CALL OUTPUT (1,0,2)
      DEFL=0.

C     BEGIN ITERATIONS IN Z

      JJ=0
   10 JJ=JJ+1
      KP=KP+1
      CALL FOLLOW
      CALL PREDCT
      CALL UPDATZ
      CALL BOUNDS(3)
      CALL INTENS
C     TEST FOR ENERGY CONSERVATION
      IF (ABSF((SUM-POWER)/POWER).GT.ECHNG) GO TO 5

      CALL OUTPUT (2,JJ,0)

C     TEST FOR RUNNING PLOT
      NOPLOT=1
      IF (Z.LT.ZPLOT(KJ)) GO TO 7
      NOPLOT=2
      CALL OUTPUT (3,JJ,3)
      GO TO 6

C     SLIP IN HERE AND GET FINAL RUNNING OUTPUT BEFORE EXITING ON ECHNG OVERF
    5 NOUT=1
      NXY=6

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

    7 CALL SHIFT (U,V,NX,NY)
    8 GO TO 10

C     Z ITERATIONS ARE FINISHED.  DO FINAL GRAPHS
    9 CALL GRAPH2

      STOP
      END
```

```
      SUBROUTINE INPUT

C     DATA CARDS READ SHOULD CONTAIN THE FOLLOWING

C     CARD 1

C        HX,HY        STEP SIZES IN THE X,Y DIRECTIONS
C        POWER        INPUT POWER IN WATTS
C        VZERO        WIND VELOCITY   CM/SEC
C                     (IN Y-DIRECTION)
C        F            FOCAL LENGTH OF FOCUSSED BEAM IN CM
C        OMEGA        SLEWING RATE IN RADIANS/SEC
C        PH2O         PARTIAL PRESSURE OF WATER IN TORR

C     CARD 2

C        NX           NUMBER OF MESH POINTS IN X DIRECTION
C                     (HALF AS MANY AS IN Y DIRECTION DUE TO SYMMETRY
C                     ABOUT WIND DIRECTION (Y-DIRECTION))
C        NY           NUMBER OF MESH POINTS IN Y DIRECTION
C        KQMAX        NUMBER OF MAX INTENSITY CALCULATIONS
C        NOLPLC       =0 IF TRANSVERSE LAPLACIAN INCLUDED
C                     =1 IF TRANSVERSE LAPLACIAN IS DROPPED
C        LPLOT        = 0  FOR GROUP PLOTTING
C                     = 1  FOR SEQUENTIAL PLOTTING
C        NFOCUS       =0 FOR FOCUSED PLOTTING
C                     =1 FOR NON-FOCUSSED PLOTTING

C     CARD 3

C        GAMMA        RATIO OF SPECIFIC HEATS
C        REFRAC       MOLECULAR REFRACTIVITY
C        C            VELOCITY OF SOUND   CM/SEC
C        ZT           DISTANCE (KM) AT WHICH INTERACTION STOPS.
C        DIAM         HALF WIDTH OF INITIAL GAUSSIAN PROFILE
C                     OR RADIUS OF APERTURE
C        THETA        ELEVATION ANGLE OF BEAM
C        WN           WAVE-NUMBER OF LIGHT SOURCE

C     CARD 4

C        RO           AMBIENT DENSITY
C        P            TOTAL AMBIENT PRESSURE IN TORR
C        ECHNG        ABSOLUTE MAGNITUDE OF RELATIVE CHANGE OF INTEGRATED
C                     INTENSITY TO CAUSE RUN ABORT
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
C        NTAU         SET =1 TO ENTER VALUE OF TAU THAT FOLLOWS.
C        TAU          V-T RELAXATION TIME
C        NALPH        SET =1 TO ENTER VALUE OF ALPHA THAT FOLLOWS.
C        ALPHA        TOTAL ABSORPTION COEFFICIENT
```

```
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

C       CARD 6

C          FORM1     A PRINTING FORMAT

C       CARD 7

C          FORM2     A PRINTING FORMAT

C       CARD 8

C          PW        MULTIPLE OF VACUUM WAIST POSITION TO GIVE WAIST
C                    POSITION
C          QW        MULTIPLE OF VACUUM WAIST TO GIVE WAIST SIZE

C       CARD 9

C          NPLOT     NUMBER OF PLOTS

C       CARDS 10...  VALUES OF Z AT WHICH PLOTS ARE DESIRED (KM)

        COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
      1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
      2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
        COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
      1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
      2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
      3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
      4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
      5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

        READ 100,HX,HY,POWER,VZERO,F,OMEGA,PH2O
100     FORMAT(7E10.3)

        READ 200,NX,NY,KQMAX,NOLPLC,LPLOT,NFOCUS
 200    FORMAT (6I5)

        READ 300,GAMMA,REFRAC,C,ZT,DIAM,THETA,WN
300     FORMAT(7E10.3)

        READ 400,RO,P,ECHNG,TEMP
  400   FORMAT(4E10.3)

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

        READ 600,(FORM1(I),I=1,2)
  600   FORMAT(A8,A6)

        READ 700,(FORM2(I),I=1,2)
  700   FORMAT(A8,A5)
```

```
      READ 800,PW,QW
 800  FORMAT(2E10.3)

      READ 900, NPLOT
900   FORMAT (I5)

      READ 1000, (ZPLOT(I),I=1,NPLOT)
1000  FORMAT (8F10.1)

      PRINT 101,HX,HY,POWER,VZERO,F,OMEGA,PH20
 101  FORMAT(* HX=*E12.5/* HY=*E12.5/* POWER=*E12.5/
     1* VZERO=*E12.5/* F=*E12.5/* OMEGA=*E12.5/* PH20=*E12.5/)
      PRINT 201,NX,NY,NPLOT,KQMAX
 201  FORMAT(* NX=*I5/* NY=*I5/* NPLOT=*I5/* KQMAX=*I5)
      IF(NOLPLC .EQ. 0) GO TO 204
      PRINT 203
 203  FORMAT(* THIS RUN HAS NO DIFFRACTION*)
 204  CONTINUE
      PRINT 205,LPLOT,NFOCUS
 205  FORMAT(* LPLOT=*I5/* NFOCUS=*I5/)
      PRINT 301,GAMMA,REFRAC,C,ZT,DIAM,THETA,WN
 301  FORMAT(* GAMMA=*E12.5/* REFRAC=*E12.5/* C=*E12.5/* ZT=*E12.5/
     1  * DIAM=*E12.5/* THETA=*E12.5/* WN=*E12.5/)
      PRINT 401,RO,P,ECHNG,TEMP
 401  FORMAT(* RO=*E12.5/* P=*E12.5/* ECHNG=*E12.5/* TEMP=*E12.5/)

      RETURN
      END
```

```
      SUBROUTINE SETUP

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

      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

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

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

      NOPLOT=1
      PI=3.1415926
      Z=0.
      ZETA=0.
      ZZ=0.
      D=1.
      DNEW=1.
      FOCUS=1.
      IQ=1
      NXY=1
      ALFSUM=0
      NOUT=0
      CZERO=C
      ROZERO=RO
      NX1=NX-1
      NY1=NY-1
      NX2 = 2*NX-2
      NY2=NY/2+1
      WIDTH=DIAM/(2.0*SQRT(2.0))
      W2=WIDTH*WIDTH
      FL=F/100000.

      KJ=1

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

      ALPHA2=.00144*((295./(273.+TEMP))**1.5)*(10.**(-970./(TEMP+273.)))
      ALPHA2=6.7E-7
      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-11)*PH2O*(P+193.*PH2O) + ALPHA2
```

```
    2 IF (NDEL.EQ.1) GO TO 3
      DELTA=(ALPHA2/ALPHA)*2.44

    3 ALFCON=ALPHA

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,NDEL,DELTA,NTAU,TAU,NALPH,ALPHA,NBETA,BETA
   12 FORMAT(* NDEL=*I5/* DELTA=*E12.5/* NTAU=*I5/* TAU=*E12.5/
     1    * NALPH=*I5/* ALPHA=*E12.5/* NBETA=*I5/* BETA=*E12.5/)

C     PRINT THE LAST INPUT ITEMS (CARD 8)
      PRINT 13,PW,QW
   13 FORMAT(* P=*E12.5,*      Q=*E12.5/)

C      CALCULATE POSITION AND SIZE OF VACUUM BEAM WAIST.
      FHAT=F/(WN*W2)
      AWV=FHAT/ SQRTF(1.+FHAT*FHAT)
      ZWV=FHAT/(1.+FHAT*FHAT)
C      CALCULATE WAIST SIZE AND POSITION OF COORDINATE SYSTEM
      AWAIST=QW*AWV
      ZWAIST=PW*ZWV
C      FACTOR = CHANGE FOR DIFF-EQ IN NEW COORDINATE SYSTEM
      FACTOR=AWAIST*AWAIST*(1.-AWAIST*AWAIST)/(ZWAIST*ZWAIST)

      ZW=ZWAIST*WN*W2
      AW=AWAIST*WIDTH
      PRINT 50,ZW,AW
50    FORMAT(*       WAIST POSITION=*E12.5 *       WAIST RADIUS=*E12.5)

      HZ=1./(BETA+4./(QW*FHAT)*(1./HX**2+(QW*PW*NX*HX)**2))
      PRINT 14,HZ
  14  FORMAT (* HZ=* F12.9)
      HZY=HZ/(HY*HY)
      HZX=HZ/(HX*HX)

C     CALCULATE NUMBER OF STEPS TO REACH LAST ZPLOT VALUE
      AW=AWAIST**2
      BB=1.-AW
      SB=SQRTF(BB)
      E=EXPF(HZ*SB/ZWAIST)
      HZKM=ZWAIST*(1.+((( .-SB)**2)*E*E-AW)/(2.*(SB-BB)*E))*WN*W2/10.**5
      NSTEP=ZPLOT(NPLOT)/HZKM + 1
      KP=KKMAX-1
      KKMAX=NSTEP/KQMAX + 1
      RETURN
      END
```

UNCLASSIFIED

```
      FUNCTION VACAMP(X,Y,N)
      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTHAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

      Z1=1.-Z*100000./F
      F2=FOCUS*FOCUS
      V1=FOCUS*EXPF(-.5*(X*X+Y*Y)*F2/D)/(D*SQRTF(PI))
      D1=(ZETA*(1.+1./(FHAT*FHAT))-1./FHAT)/D
      DN1=(ZETA/ZWAIST-1.)*(1.-AWAIST*AWAIST)/(ZWAIST*DNEW)
      ALPH = 0.5*(X*X + Y*Y)*(D1 - DN1)*F2
      C1=COSF(ALPH )
      S1=SINF(ALPH )
      GO TO (1,2) N
1     CONTINUE
      VACAMP=V1*(Z1*C1+ZETA*S1)
      GO TO 100
2     CONTINUE
      VACAMP=-V1*(ZETA*C1-Z1*S1)
100   CONTINUE
      END
```

```
      SUBROUTINE INITL (N)

      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

      GO TO (10,20) N

   10 CONTINUE
      X=-HX
      DO 11 I=1,NX1
      X=X+HX
      Y=-(NY2-1)*HY
      DO 11 J=2,NY1
      Y=Y+HY
      U(I,J,1)=VACAMP(X,Y,1)
      V(I,J,1)=VACAMP(X,Y,2)
   11 CONTINUE
      CALL BOUNDS(1)
      RETURN

   20 CONTINUE
      CALL BOUNDS(2)
      Y=-(NY2-1)*HY

C      THIS PART IS SEPARATE SINCE SYMMETRY MAKES I=1 A SPECIAL LINE.

      DO 24 J=2,NY1
      X=0.
      Y=Y+HY
      IF (NOLPLC.EQ.1) GO TO 21

C      FORM DIFFERENCES FOR DERIVATIVES

      DELUX=(U(2,J,1)-U(1,J,1))*2.0
      DELVX=(V(2,J,1)-V(1,J,1))*2.0
      DELUY=U(1,J+1,1)-2.*U(1,J,1)+U(1,J-1,1)
      DELVY=V(1,J+1,1)-2.*V(1,J,1)+V(1,J-1,1)

C      PREDICTED VALUES

      XYD=(X*X+Y*Y)/FOCUS
      XYD=XYD*FACTOR
      U(1,J,2)=U(1,J,1)-0.5*HZX*DELVX-0.5*HZY*DELVY+(XYD     -CC(1,J))
     1*V(1,J,1)*0.5*HZ
      V(1,J,2)=V(1,J,1)+0.5*HZX*DELUX+0.5*HZY*DELUY-(XYD     -CC(1,J))
     1*U(1,J,1)*0.5*HZ
      GO TO 22
```

```
C      REPLACE LAPLACIAN METHOD WITH THE FOLLOWING WHEN NOLPLC=1
    21 U(I,J,2)=U(I,J,1)-CC(1,J)*V(I,J,1)*HZ*0.5
       V(I,J,2)=V(I,J,1)+CC(1,J)*U(I,J,1)*HZ*0.5
22     CONTINUE
       DO 24 I=2,NX1
       X=X+HX
       IF(NOLPLC.EQ.1) GO TO 23
C       FORM DIFFERENCES FOR DERIVATIVES
       DELUX=U(I+1,J,1)-2.*U(I,J,1)+U(I-1,J,1)
       DELVX=V(I+1,J,1)-2.*V(I,J,1)+V(I-1,J,1)
       DELUY=U(I,J+1,1)-2.*U(I,J,1)+U(I,J-1,1)
       DELVY=V(I,J+1,1)-2.*V(I,J,1)+V(I,J-1,1)
C       PREDICTED VALUES
       XYD=(X*X+Y*Y)/FOCUS
       XYD=XYD*FACTOR
       U(I,J,2)=U(I,J,1)-0.5*HZX*DELVX-0.5*HZY*DELVY+(XYD    -CC(I,J))
      1*V(I,J,1)*0.5*HZ
       V(I,J,2)=V(I,J,1)+0.5*HZX*DELUX+0.5*HZY*DELUY-(XYD    -CC(I,J))
      1*U(I,J,1)*0.5*HZ
       GO TO 24
C      REPLACE LAPLACIAN METHOD WITH THE FOLLOWING WHEN NOLPLC=1
    23 U(I,J,2)=U(I,J,1)-CC(I,J)*V(I,J,1)*HZ*0.5
       V(I,J,2)=V(I,J,1)+CC(I,J)*U(I,J,1)*HZ*0.5
    24 CONTINUE
    25 RETURN
       END
```

```
      SUBROUTINE PREDCT
C***********************************************************************
C      THIS ROUTINE IS A TWO-POINT PREDICTOR SOLVING THE APPROXIMATE
C      WAVE EQUATION.
C***********************************************************************

      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

C      GET CURRENT INDEX CHANGES INTO THE ARRAY CC(I,J)
C      SKIP INDEX CALL FOR VACUUM RUN

      IF (BETA.EQ.0) GO TO 1

      CALL INDEX(2)
1     CONTINUE
      Y=-HY*(NY2-1) + DEF
      DO 5 J=2,NY1
      X=0.
      Y=Y+HY

C      USE TRANSVERSE LAPLACIAN UNLESS NOLPLC=1
      IF (NOLPLC.EQ.1) GO TO 2

C      FORM DIFFERENCES FOR DERIVATIVES

      DELVX=(V(2,J,2)-V(1,J,2))*2.0
      DELUX=(U(2,J,2)-U(1,J,2))*2.0
      DELUY=U(1,J+1,2)-2.*U(1,J,2)+U(1,J-1,2)
      DELVY=V(1,J+1,2)-2.*V(1,J,2)+V(1,J-1,2)

C      PREDICTED VALUES

      XYD=(X*X+Y*Y)/FOCUS
      XYD=XYD*FACTOR
      U(1,J,3)=U(1,J,1)-HZX*DELVX-HZY*DELVY+(XYD-CC(1,J))*V(1,J,2)*HZ
      V(1,J,3)=V(1,J,1)+HZX*DELUX+HZY*DELUY-(XYD-CC(1,J))*U(1,J,2)*HZ
      GO TO 3

C      DROP TRANSVERSE LAPLACIAN WHEN NOLPLC=1

    2 U(1,J,3)=U(1,J,1)-CC(1,J)*V(1,J,2)*HZ
      V(1,J,3)=V(1,J,1)+CC(1,J)*U(1,J,2)*HZ
```

```
    3     CONTINUE
          DO 5 I=2,NX1
          X=X+HX
C         USE TRANSVERSE LAPLACIAN UNLESS NOLPLC=1
          IF (NOLPLC.EQ.1) GO TO 4
C          FORM DIFFERENCES FOR DERIVATIVES
          DELUX=U(I+1,J,2)-2.*U(I,J,2)+U(I-1,J,2)
          DELVX=V(I+1,J,2)-2.*V(I,J,2)+V(I-1,J,2)
          DELUY=U(I,J+1,2)-2.*U(I,J,2)+U(I,J-1,2)
          DELVY=V(I,J+1,2)-2.*V(I,J,2)+V(I,J-1,2)
C          PREDICTED VALUES
          XYD=(X*X+Y*Y)/FOCUS
          XYD=XYD*FACTOR
          V(I,J,3)=V(I,J,1)+HZX*DELUX+HZY*DELUY-(XYD-CC(I,J))*U(I,J,2)*HZ
          U(I,J,3)=U(I,J,1)-HZX*DELVX-HZY*DELVY+(XYD-CC(I,J))*V(I,J,2)*HZ
          GO TO 5
C         DROP TRANSVERSE LAPLACIAN WHEN NOLPLC=1
        4 U(I,J,3)=U(I,J,1)-CC(I,J)*V(I,J,2)*HZ
          V(I,J,3)=V(I,J,1)+CC(I,J)*U(I,J,2)*HZ
        5 CONTINUE
          RETURN
          END
```

```
      SUBROUTINE BOUNDS(KK)
C       ASSIGN BOUNDARY VALUES
      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ
      DO 1 I=1,NX
      U(I,NY,KK)=0.
    1 V(I,NY,KK)=0.
      DO 2 I=1,NX
      U(I,1,KK)=0.
    2 V(I,1,KK)=0.
      DO 3 J=1,NY
      U(NX,J,KK)=0.
    3 V(NX,J,KK)=0.
      RETURN
      END
```

```
      SUBROUTINE SHIFT
      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ
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)
      U(I,J,2)=U(I,J,3)
      V(I,J,1)=V(I,J,2)
    1 V(I,J,2) = V(I,J,3)
      RETURN
      END
```

```
      SUBROUTINE FOLLOW
      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

      JCENM=NY2
      CMAX=0.
      DO 1 J=1,NY
      IF(QI(J,NX).LE.CMAX) GO TO 1
      CMAX=QI(J,NX)
      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
      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
      GO TO 15
13    CONTINUE
      DEFL=DEFL+HY*WIDTH*FOCUS
      DEF=DEFL/(WIDTH*FOCUS)
      GO TO 14
15    CONTINUE
      DEFL=DEFL-HY*WIDTH*FOCUS
      DEF=DEFL/(WIDTH*FOCUS)
14    CONTINUE
      PRINT 400,DEFL,Z
400   FORMAT(* DEFLECTION=*,E12.5,*CM AT Z =*,F10.3,* KM*)
      CALL BOUNDS(1)
      CALL BOUNDS(2)
12    CONTINUE
      RETURN
      END
```

```
      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(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

      DIMENSION A(61),B(61)

      ARG=WIDTH*FOCUS/(TAU*(VZERO+OMEGA*ZETA*W2*WN))
      CONST=BETA*EXPF(-ALFSUM          )/(1.+OMEGA*WN*W2*ZETA/VZERO)
      EX=EXPF(-ARG*HY)
      DO 1 I=1,NX
      CC(I,1)=0.
      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)
      CC(I,2)=A(2)+B(2)
      DO 1 J=3,NY
      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)
1     CONTINUE

      RETURN
      END
```

```
      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(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

      DIMENSION AREA(9), CONMIN(9), SUMA(9)
      DATA (IGATE=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 101 I=1,NX
      NN=NX+1-I
      QI(J,I)=U(NN,J,1)*U(NN,J,1)+V(NN,J,1)*V(NN,J,1)
  101 IF(QI(J,I) .GT. QMAX) QMAX=QI(J,I)
      DO 102 I=1,NX1
102   QI(J,I+NX)=QI(J,NX-I)
      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.NORPLOT.EQ.2) IGATE=3

C      FACTOR FOR SHRINKING COORDINATE SYSTEM
      F2=DNEW
      DO 3 J=1,NY
      DO 2 I=1,NX
      NN=NX+1-I
    2 QI(J,I)=(U(NN,J,3)*U(NN,J,3)+V(NN,J,3)*V(NN,J,3))
      DO 3 I=1,NX1
3     QI(J,I+NX)=QI(J,NX-I)

      GO TO (7,7,4) IGATE

C     CALCULATE MAXIMUM INTENSITIES FOR PLOTTING
    4 QMAX=QCMAX=0
      DO 6 J=2,NY1
      IF (QI(J,NX).LE.QCMAX) GO TO 5
      QCMAX = QI(J,NX)
```

```
      JCMAX=J
    5 DO 6 I=2,NX
      IF(QI(J,I) .LE. QMAX) GO TO 6
      QMAX=QI(J,I)
      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
      TEMP1=HX*HY
      DO 12 I=2,NX2
      DO 12 J=2,NY1
      TEMP2=POWER*QI(I,J)*TEMP1
      SUM=SUM+TEMP2
      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
   10 AREA(L)=AREA(L)+TEMP1
      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(-ALFSUM            )
      II=II+1
      ZI(II)=Z
   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
      AREA(I)=AREA(I)*W2*F2
      AVGINT=SUMA(I)/AREA(I)
      VACINT=POWER*(1.-CONMIN(I))/(PI*W2*LOGF(1./CONMIN(I))*D)*
     1   EXP(-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(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

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(31), DENS(31,61), NQI(31)

      GO TO (10,20,30,40), II

C      PRINT AMPLITUDES

   10 J1=1
11    CONTINUE
      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 12,Z
   12 FORMAT(8H1U AT Z=E12.5)
      PRINT FORM2,((U(I,J,KK),I=1,NX1,2),J=J1,NY)
      PRINT 15,Z
15    FORMAT(*1 UANALYTIC AT Z=*E12.5)
      GO TO 2000
19    CONTINUE
```

UNCLASSIFIED

```
      PRINT 13,Z
   13 FORMAT(8H1V AT Z=E12.5)
      PRINT FORM2,((V(I,J,KK),I=1,NX1,2),J=J1,NY)
      PRINT 16,Z
16    FORMAT(*1 V ANALYTIC AT Z=*E12.5)
2000  CONTINUE
      PRINT FORM2,((DENS(J,K),J=1,NX1,2),K=1,NY)
17    CONTINUE
      RETURN

C     CALCULATE, PRINT AND STORE DATA FOR TERMINAL PLOTS EVERY KKMAXTH INTE

20    CONTINUE
   21 IF (KP.LT.KKMAX.AND.NOPLOT.EQ.1) GO TO 28
      KP=0
      DTEMP=D
      D=DNEW
      PRINT 23,SUM,Z
   23 FORMAT(22H INTEGRATED INTENSITY=E15.8,12H WATTS AT Z=E12.5,
     111H KILOMETERS)
      SUM2 = ABSF((SUM-SUM1)/SUM1)
      PRINT 24, SUM2
   24 FORMAT(1X,*DELTA-E/E=*E12.5)

C      MAXIMUM INTENSITY CALCULATION

      QP=QI(JMAX,IMAX)-QI(JMAX,IMAX+1)
      QM=QI(JMAX,IMAX)-QI(JMAX,IMAX-1)
      QQMAX(IQ)=QI(JMAX,IMAX)+0.125*(QP-QM)**2/(QP+QM)
      QQMAX(IQ)=QQMAX(IQ)*POWER*EXPF(-ALFSUM)/(W2*D)
      ZM(IQ)=Z
      PRINT 25, QQMAX(IQ),ZM(IQ),IMAX,JMAX,DEFL
   25 FORMAT(16H PEAK INTENSITY=E12.5,6H AT Z=E12.5/
     16H IMAX=I5/6H JMAX=I5/17H BEAM DEFLECTION=E12.5/)
      QP=QI(JCMAX,NX)-QI(JCMAX+1,NX)
      QM=QI(JCMAX,NX)-QI(JCMAX-1,NX)
      SJMAX=JCMAX+(QM-QP)/(2.*(QM+QP))
      CBD=((SJMAX-1.)*HY-(NY2-1)*HY)*SQRTF(D)*WIDTH
      QJMAX(IQ)=CBD+DEFL
      QQQMAX=QI(JCMAX,NX)+0.125*(QP-QM)**2/(QP+QM)
      QQQMAX=QQQMAX*POWER*EXPF(-ALFSUM)/(W2*D)
      PRINT 26,QQQMAX,Z,CBD
   26 FORMAT(25H CENTRAL PEAK INTENSITY =E12.5,6H AT Z=E12.5/
     125H CENTRAL BEAM DEFLECTION=E12.5/)

C      CALCULATE GEBHARDT AND SMITH N-PARAMETER AND I-REL
      ZALF=ALFSUM
      GSN=BETA*(1.-(1.-EXPF(-ZALF))/ZALF)*ZETA/(  PI*ALFCON*WN*W2)
      AZSQ=W2*D
      RELI=QQQMAX*PI*AZSQ/(POWER*EXPF(-ZALF))
      PLTGSN(IQ)=GSN
      PLTREL(IQ)=RELI
      IQ=IQ+1
      PRINT 27,GSN,RELI,Z
   27 FORMAT(* N=*F10.5,* RELI=*F10.5,* AT Z=*E12.5* KILOMETERS*)
      D=DTEMP
```

```
   28 RETURN

C     CALCULATE AND PRINT RELATIVE DENSITY CHANGES (USE KK=3)

30    ZZM=Z
      PRINT 31, ZZM
   31 FORMAT(*1RELATIVE DENSITY CHANGES TIMES 100 AT Z=*E12.5* KILOMETER
     1S*)
      DO 32 I=1,NX
      DO 32 J=1,NY
   32 DENS(I,J)=(CC(I,J)/(3.*REFRAC*W2*WN*WN))/(RO*.01)
      PRINT FORM2,((DENS(I,J),I=1,NX1,2),J=2,NY1)

C     CALCULATE AND PRINT PHASES

33    PRINT 34,Z
   34 FORMAT(13H1PHASES AT Z=E12.5,11H KILOMETERS)
      DO 37 J=2,NY1
      DO 36 I=1,NX1,2
      IF(U(I,J,3).EQ.0..AND.V(I,J,3).EQ.0.) GO TO 35
      ARCT(I)=ATAN2(V(I,J,3),U(I,J,3))
      GO TO 36
   35 ARCT(I)=0.
   36 CONTINUE
      PRINT FORM2,(ARCT(I),I=1,NX1,2)
   37 CONTINUE

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

      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
```

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

      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ

      DIMENSION IMAGE(61,61),QQ(61),XQ(61)

C      PLOT EQUI-INTENSITY CONTOURS IN X,Y SPACE

      FF=FOCUS
      IF(NFOCUS.EQ.1) FF=1
      XIN=8.0*FF
      YIN=8.0*FF

      YMIN=-(NX-1)*HX*WIDTH*FOCUS/FF
      XMIN=YMIN+DEFL
      H1Y=H1X=ABSF(YMIN)/5.
      DD=DNEW
      CALL TOPOGRAF(QI,61,61,NX2,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,DD,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(J,NX)*POWER*EXPF(-ALFSUM)/(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,QQ,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(12.,0,-3)

3     IF(NOUT.EQ.1) PRINT 4,Z
    4 FORMAT (38H ENERGY CHANGES TOO LARGE, ABORT AT Z= E12.5,3H KM//)

C     INCREMENT ZPLOT INDEX
      KJ=KJ+1

      RETURN
      END
```

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

      COMMON /1/ CC(31,61), FORM1(2), FORM2(2), PLTGSN(200),
     1 PLTRAT(200,3), PLTREL(200), QI(61,61), QJMAX(200), QQMAX(200),
     2 U(31,61,3),V(31,61,3),ZI(200),ZM(200),ZPLOT(20)
      COMMON /SINGLS/ALFCON,ALFSUM,ALPHA,AWAIST,BETA,C,CZERO,D,DEF,DEFL,
     1 DELTA,DIAM,DNEW,ECHNG,F,FACTOR,FHAT,FL,FOCUS,GAMMA,HX,HY,HZ,HZX,
     2 HZY,IMAX,IQ,JCMAX,JMAX,KJ,KKMAX,KP,KQMAX,LPLOT,NALPH,NBETA,NDEL,
     3 NFOCUS,NOPLOT,NOUT,NOLPLC,NPLOT,NTAU,NX,NXY,NX1,NX2,NY,NY1,NY2,
     4 OMEGA,P,PH2O,PI,PLTCAL,POWER,PW,QMAX,QW,REFRAC,RO,ROZERO,SUM,SUM1
     5 ,TAU,TEMP,THETA,VZERO,W2,WIDTH,WN,Z,ZETA,ZT,ZWAIST,ZZ
      INTEGER XFORM,YFORM
C      PLOT MAXIMUM INTENSITY VS RANGE
      IQQ=IQ-1
      QQMAX(IQ)=0.0
      CALL SCALE(QQMAX,IQ   , 7.0,YMIN,DY,1,TKY)
      CALL SCALE(ZM,IQ   ,10.0,XMIN,DX,1,TKX)
      XFORM=IFORMAT (XMIN,DX,10.,TKX)
      YFORM=IFORMAT (YMIN,DY,7.,TKY)
      CALL LINE(ZM,QQMAX,IQQ,1,-1,.035,0)
      CALL AXIS(0,0,24HDISTANCE FROM LASER FACE,-24,10.0,0.0,TKX,
     1 XMIN,DX,XFORM)
      CALL AXIS(0,0,17HMAXIMUM INTENSITY,17, 7.0,90.0,TKY,YMIN,DY,
     1 YFORM)
      CALL PLOT(12.,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,IQ, 7.0,YMIN,DY,1,TKY)
      YFORM=IFORMAT (YMIN,DY,7.,TKY)
      CALL LINE(ZM,QJMAX,IQQ,1,-1,.035,0)
      CALL AXIS(0,0,24HDISTANCE FROM LASER FACE,-24,10.0,0.0,TKX,
     1 XMIN,DX,XFORM)
      CALL AXIS(0,0,15HBEAM DEFLECTION,15, 7.0,90.0,TKY,YMIN,DY,YFORM)
      CALL PLOT(12.,0,-3)
2      CONTINUE
C      PLOT RATIO OF INTENSITY TO VAC.INT.--VS RANGE
      CALL SCALE(ZI,IQQ,10.,XMIN,DX,1,TKX)
      CALL SCALE(PLTRAT(1),400+IQ,7.0,YMIN,DY,1,TKY)
      XFORM=IFORMAT (XMIN,DX,10.,TKX)
      YFORM=IFORMAT (YMIN,DY,7.,TKY)
      CALL LINE(ZI,PLTRAT(1),IQQ,1,0,.08,1)
      CALL LINE(ZI,PLTRAT(201),IQQ,1,5,.08,1)
      CALL LINE(ZI,PLTRAT(401),IQQ,1,4,.08,1)
      CALL AXIS(0,0,24HDISTANCE FROM LASER FACE,-24,10.0,0.0,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(12.,0,-3)

C       PLOT RELATIVE INTENSITY VS GEBHARDT AND SMITH N-PARAMETER
C       BUT ONLY IF THIS IS A NON VACUUM RUN

      IF(BETA.EQ.0.0) GO TO 100
      IQQ=IQ-1
      PLTREL(IQ)=0.0
      CALL SCALE(PLTGSN,IQ  ,10.0,XMIN,DX,1,TKX)
      CALL SCALE(PLTREL,IQ  , 7.0,YMIN,DY,1,TKY)
      XFORM=IFORMAT (XMIN,DX,10.,TKX)
      YFORM=IFORMAT (YMIN,DY,7.,TKY)
      CALL LINE(PLTGSN,PLTREL,IQQ,1,-1,.035,0)
      CALL AXIS(0,0,11HN-PARAMETER,-11,10.0,0.0,TKX,XMIN,DX,XFORM)
      CALL AXIS(0,0,18HRELATIVE INTENSITY,18, 7.0,90.0,TKY,YMIN,DY,
     1 YFORM)
      CALL PLOT(12.,0,-3)

100   CONTINUE

      CALL STOPPLOT

      RETURN
      END
```

```
      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 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,PH2O,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=8.0
      YI=8.0
      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 - DF

      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
      CALL AXIS(0,0,YLABEL,NCY,YI,90.,.8,YMIN,DYT,YFORMAT)
      CALL AXIS(0,0,XLABEL,NCX,XI,0.0,.8,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,.105,5HPH2O=,0.0,5)
      CALL NUMBER(1.7,7.25,.105,PH2O,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)
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(12.0,0,-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 (CX0, ICX0), (CY0,ICY0), (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
      XSHIFT=XOFF(NXY)+4.0-XINCHES/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
      IF (IMAGE(IX,IY-1).EQ.0) GO TO 8
      CY0 = (IY-1-(T(IX,IY)-FLEVEL)/(T(IX,IY)-T(IX,IY-1)))*YFACTOR
      GO TO 7
```

```
    6   CY0 = 0
    7   CX0 = (IX-1)*XFACTOR
        CX0=CX0+XSHIFT
        CY0=CY0+YSHIFT
        CALL PLOT (CX0,CY0,3)
        INOUT = 2
        GO TO 20

C       START AN INNER BOUNDARY
    8   INOUT = 1
        CX0 = (IX-2)*XFACTOR
        CY0=(IY-2+(T(IX-1,IY-1)-FLEVEL)/(T(IX-1,IY-1)-T(IX-1,IY)))*YFACTOR
        CX0=CX0+XSHIFT
        CY0=CY0+YSHIFT
        CALL PLOT (CX0,CY0,3)
        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.CX0) GO TO 16
        YGAP = ABSF(CY - CY0)
        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
```

```
      SUBROUTINE SCALE (Y,N,HEIGHT,YMIN,DY,K,TICK)
C     TIKSCALE WILL CONVERT THE N VALUES IN ARRAY Y (OF DIMENSION N*K),
C     TO INCHES, WHERE HEIGHT IS THE MAXIMUM NUMBER OF INCHES.
C     YMIN,DY,TICK, AND IFORMAT ARE PROVIDED BY TIKSCALE FOR USE BY AXIS
C        YMIN = THE DATA VALUE AT 0 INCHES,
C        DY = THE DATA INCREMENT / TICK, AND
C        TICK = DISTANCE BETWEEN TICKS (INCHES).
C        IFORMAT = A FORMAT FOR AXIS LABELING TO FIT THE DATA SCALED
C     TIKZERO INCLUDES ZERO AMONG THE VALUES TO BE SCALED BY TIKSCALE.

      DIMENSION Y(N)
      YMIN = YMAX = Y(1)
      GO TO 1

      ENTRY TIKZERO
      YMIN = YMAX = 0

    1 M = N*K
      DO 5 I=1,M,K
      IF (Y(I) - YMIN) 2,5,3
    2 YMIN = Y(I)
      GO TO 5
    3 IF (YMAX - Y(I)) 4,5,5
    4 YMAX = Y(I)
    5 CONTINUE
      IF (YMAX - YMIN) 6,6,7

C     SCALE A CONSTANT ARRAY BETWEEN 0 AND 1 INCH
    6 DY = TICK = 1.0
      IF (YMIN.EQ.0.) GO TO 10
      YMIN = 0
      DYINCH = DY = ROUNDEXP(YMAX)*10./HEIGHT
      CALL GETTICK (DY,TICK)
      GO TO 8

C     TRUNCATE YMIN TO NEAREST ROUND NUMBER
    7 YMIN = ROUNDOWN(YMIN)
      DYINCH = DY = (ROUNDUP(YMAX) - YMIN)/HEIGHT
      CALL GETTICK(DY,TICK)
      X = YMIN/DY
      XT = I = X
      IF (X.EQ.XT) GO TO 8
      YMIN = REDUCE (YMIN)
      GO TO 7

C     SCALE ARRAY WITH YMIN AND DY

    8 DO 9 I=1,M,K
    9 Y(I) = (Y(I) - YMIN)/DYINCH

C     SELECT APPROPRIATE FORMAT FOR THIS DATA

   10 ENDTICK=YMIN+DY*INTF(HEIGHT/TICK)
      BIG=MAX1F(ABS(YMIN),ABS(ENDTICK))
      SMALL=DY
      CALL NORMAL (SMALL,IEXP)
      CALL NORMAL (BIG,NEXP)
      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
```

UNCLASSIFIED

```
      SUBROUTINE GETTICK (DY,TICK)
C     GETTICK TAKES ANY GIVEN NUMBER OF UNITS PER INCH, DY, FROM SCALE AND
C     GIVES BACK TWO NEW VALUES, DY AND TICK, FOR USE IN AXIS, WHERE THE NEW
C     DY = THE NUMBER OF UNITS PER TICK, AND TICK IS BETWEEN .8 AND 2 INCHES
C     THE NEW DY = 1,2,OR 5 TIMES SOME POWER OF 10.
      D = ABSF(DY)
      CALL NORMAL(D,IEXP)
      IF (D-5.0) 2,1,4
    1 TICK = 1.0
      RETURN
    2 IF (D-2.5) 3,3,5
    3 IF (D-1.0) 7,1,6
    4 DY = 10.*10.**IEXP
      TICK = 10./D
      RETURN
    5 DY = 5.*10.**IEXP
      TICK = 5.0/D
      RETURN
    6 DY = 2.*10.**IEXP
      TICK = 2.0/D
      RETURN
    7 PRINT 100, DY
  100 FORMAT  (/30H *****ERROR IN GETTICK -- DY = E10.3/)
      RETURN
      END
```

```
      FUNCTION ROUND (X)
C     FUNCTION ROUND GIVES THE ROUNDED VALUE OF ANY + OR - REAL NUMBER.
C     THE RESULT IS AN INTEGER TIMES SOME POWER OF TEN.
C     EXAMPLES OF THE THREE ROUND FUNCTIONS
C         X          ROUND(X)   ROUNDUP    ROUNDOWN
C         .........  .........  .........  .........
C         534.0      500.0      600.0      500.0
C         0.6666     0.7        0.7        0.6
C         -1.5       -2.0       -1.0       -2.0
C         -.0333     -.03       -.03       -.04
      ARG = X
      CALL NORMAL (ARG,IEXP)
      IF (ARG) 3,2,1
    1 TARG = INTF(ARG+.5)
      GO TO 4
    3 TARG = INTF(ARG-.5)
    4 ROUND = TARG*10.**IEXP
    2 RETURN
C     ROUNDOWN GIVES THE NEXT LOWEST ROUND NUMBER
      ENTRY ROUNDOWN
      ARG = X
      CALL NORMAL (ARG,IEXP)
      TARG = INTF(ARG)
      IF (TARG.GT.ARG) TARG = TARG - 1.0
      ROUND = TARG*10.**IEXP
      RETURN
C     ROUNDUP GIVES THE NEXT HIGHEST ROUND NUMBER
      ENTRY ROUNDUP
      ARG = X
      CALL NORMAL (ARG,IEXP)
      TARG = INTF(ARG)
      IF (TARG.LT.ARG) TARG=TARG+1.0
      ROUND = TARG*10.**IEXP
      RETURN
C     ROUNDEXP(X) GIVES THE SCALE FACTOR FOR A GIVEN X, WHEN REDUCED TO THE
C     NORMALIZED FORM, BETWEEN 1.000 AND 9.999 TIMES THE SCALE FACTOR.
C     WHEN X=0, ROUNDEXP GIVES THE SCALE FACTOR FOR THE PREVIOUSLY ROUNDED
      ENTRY ROUNDEXP
      IF (X.EQ.0) GO TO 5
      ARG = X
      CALL NORMAL (ARG,IEXP)
      GO TO 6
    5 IF (ABSF(TARG).EQ.10.) IEXP = IEXP+1
    6 ROUND = 10.**IEXP
      RETURN
      END
      FUNCTION REDUCE (X)
C     REDUCE TAKES A NUMBER X = A*10**EXP, WHERE 1.LE.A.LT.10,
C     AND REDUCES IT BY 1*10**EXP.   REDUCE(X) = (A-1)*10**EXP
C     SIMILARLY, AUGMENT(X) = (A+1)*10**EXP
      ARG = X
      CALL NORMAL (ARG,IEXP)
      REDUCE = (ARG-1.)*10.**IEXP
      RETURN
      ENTRY AUGMENT
      ARG = X
      CALL NORMAL (ARG,IEXP)
      REDUCE = (ARG+1.)*10.**IEXP
      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
```