

12

**SSPARAMA: A Nonlinear, Wave Optics
Multipulse (and CW) Steady-State Propagation
Code with Adaptive Coordinates**

K. G. WHITNEY AND G. L. MADER

*Science Applications Inc.
Arlington, Virginia*

and

P. B. ULRICH

*Optical Radiation Branch
Optical Sciences Division*

February 10, 1977

PERMIT POLY REPRODUCED



DDC
RECEIVED
MAR 22 1977
D

**NAVAL RESEARCH LABORATORY
Washington, D.C.**

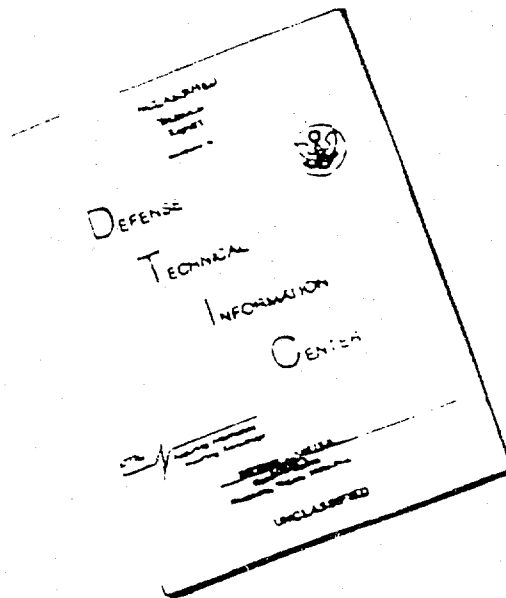
Approved for public release; distribution unlimited.

*This reproduction contains some illegible pages. It is the
best that can be obtained from the copy furnished to DDC.*

ADA037395

DDC FILE COPY

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER NRL 8074	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) SSPARAM: A NONLINEAR, WAVE OPTICS, MULTIPULSE (AND CW) STEADY-STATE PROPAGATION CODE WITH ADAPTIVE COORDINATES	5. TYPE OF REPORT & PERIOD COVERED Final report - one phase of a continuing NRL problem		
6. AUTHOR(s) K. G. Whitney, G. L. Mader, and P. B. Ulrich	7. PERFORMING ORG. REPORT NUMBER		
8. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375	9. CONTRACT OR GRANT NUMBER(s) SAA-76		
10. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command (PMG-405) Washington, D.C. 20362	11. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem R05-91 Project SAA-76		
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. REPORT DATE February 10, 1977		
	14. NUMBER OF PAGES 80		
	15. SECURITY CLASS. (for the report) Unclassified		
	16. DECLASSIFICATION/DOWNGRADING SCHEDULE		
17. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited			
18. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
19. SUPPLEMENTARY NOTES			
20. KEY WORDS (Continue on reverse side if necessary and identify by block number) Laser propagation Nonlinear optics Atmospheric optics Computer simulation			
21. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the numerical procedures that are used to calculate the steady-state atmospheric propagation and thermal blooming of either a train of high-energy laser pulses or a CW beam. The calculation is performed through the aid of a sequence of coordinate and variable transformations that enable one to adapt the scale and location of the transverse coordinate system to the amount of diffraction or thermal blooming taking place during beam propagation. A description is also given of the input parameters and diagnostics provided in the calculation along with a listing of the computer program.			

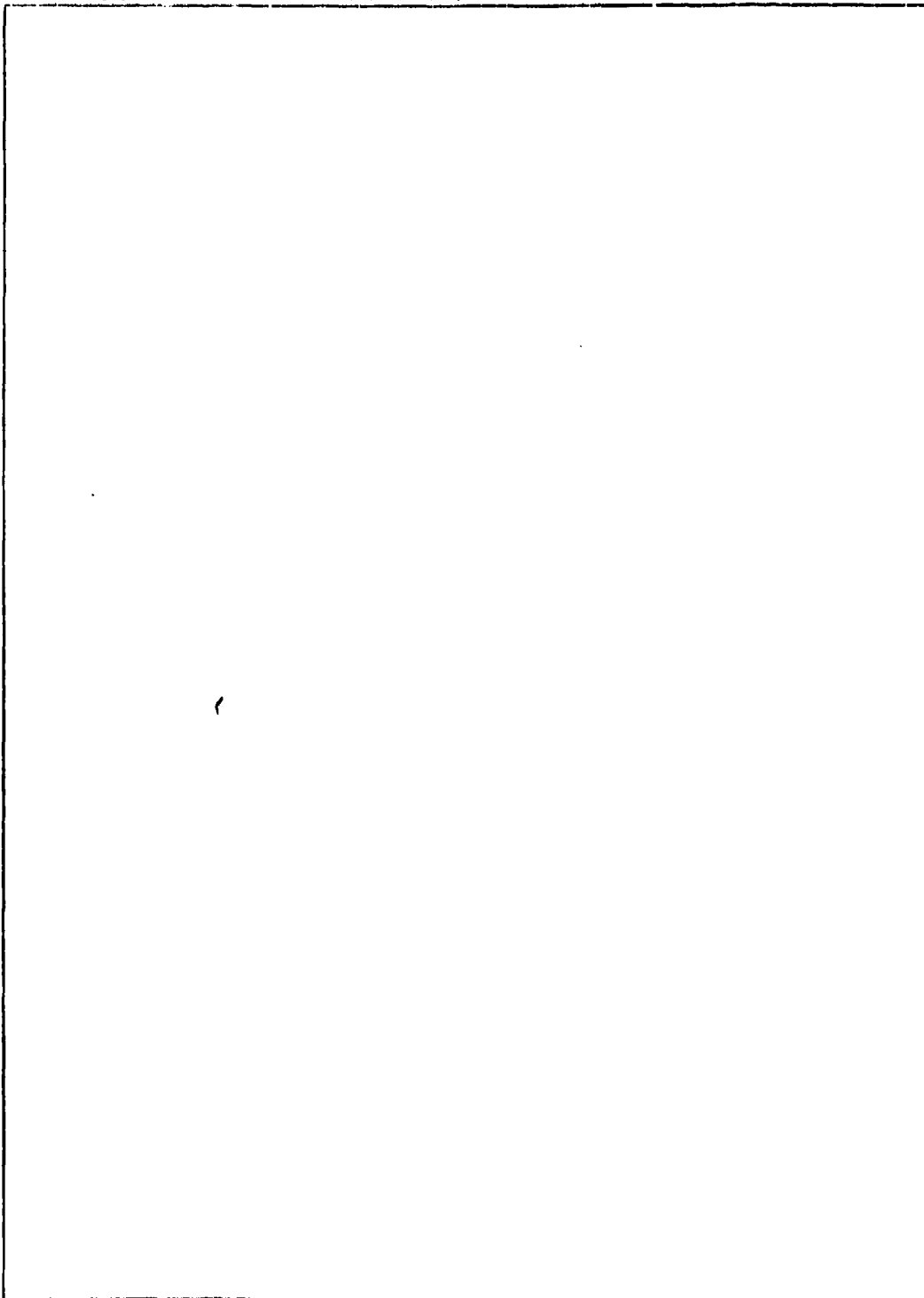
DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

LB

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



II SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

	Page
INTRODUCTION	1
COORDINATE-SYSTEM ADAPTION	4
NUMERICAL PROCEDURES	11
PROGRAM OPERATION	14
First Input Card	15
Second Input Card	18
Examples of Output	19
Summary of Program Structure	23
REFERENCES	26
APPENDIX A—Listing of Code and Comments	27

ADDITION
Dr

178

179

180

White Section ☒

Diff Section ☐

☐

LITY 0003

SPECIAL

A 23

076

D D C

RECEIVED

MAR 22 1977

RECEIVED

D

SSPARAMA: A NONLINEAR, WAVE OPTICS MULTIPULSE (AND CW) STEADY-STATE PROPAGATION CODE WITH ADAPTIVE COORDINATES

INTRODUCTION

Several methods of propagating CW high-energy laser beams through the atmosphere have been reported previously [1,2]. This report will describe a method for propagating multiply pulsed laser beams in a nonlinear atmosphere by adapting the coordinate system to the amount of thermal blooming. This technique increases the accuracy of thermal blooming calculations and extends the capability of the code in the case of extreme beam distortion.

The computer code SSPARAMA calculates the steady-state intensity pattern of a train of high-energy laser pulses propagating through the atmosphere in the presence of thermal blooming. Steady state is achieved when enough equally spaced, equal-energy pulses have been propagated for transients in air heating to have died out. In the steady state a single pulse will propagate in an atmosphere that has been heated by many preceding pulses which have the same energy distribution as the pulse one is calculating. The pulse widths are assumed to be short compared to the sound transit time across the face of the beam, so that self-blooming will not take place. Blooming occurs only as a result of air heating by preceding pulses. However, to avoid problems of plasma formation, the pulse width must be sufficiently long that the critical intensity for air breakdown is not exceeded. Finally, as the pulse is propagated from one coordinate plane to another, coordinate transformations are performed to insure that the transverse scale lengths are adapted to the amount of thermal blooming induced on the pulse train by the negative lensing influence of the heated atmosphere.

Another requirement for steady-state propagation is that a cooling mechanism exist for removing heated air from the path of the beam. In SSPARAMA, cooling is provided either by a wind moving perpendicular to the propagation direction or by beam sluing about an axis in the aperture plane perpendicular to both the wind and the propagation directions. The steady-state density changes $\Delta\rho$ introduced in the path of a given pulse by energy absorption from all preceding pulses can then be expressed as [3]

$$\Delta\rho = -\frac{\gamma-1}{c_s^2} \alpha E_p e^{-\alpha z} \sum_{n=1}^{\infty} \left| \phi(x - n\Delta t_s(v_0 + \Omega z), y, z) \right|^2, \quad (1)$$

where

- z = the distance in the propagation direction measured from the aperture plane,
- x = the distance in the wind direction measured from beam maximum intensity in the aperture plane,
- γ = the ratio of atmospheric specific heats (≈ 1.4),
- c_s = the speed of sound in air (≈ 340 m/s),
- α = the absorption coefficient for the laser radiation,
- Δt_s = the pulse spacing,
- E_p = the energy of each laser pulse,
- v_0 = the wind speed along the x direction perpendicular to the direction of propagation, and
- Ω = the angular sluing rate of the beam about the y axis.

Finally ϕ is the normalized steady-state energy distribution of each pulse at the z plane:

$$\int_{-\infty}^{\infty} |\phi(x, y, z)|^2 dx dy = 1. \quad (2)$$

This density reduction $\Delta\rho$ changes the index of refraction from its ambient value n_0 , where $n_0 \approx 1$, to

$$n^2 \approx n_0^2 + 3N\Delta\rho,$$

where N is the molecular refractivity of air (≈ 0.154 cm³/g). The distribution ϕ must then be calculated self-consistently from the propagation equation:

$$\left[2ik \frac{\partial}{\partial z} + \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + 3Nk^2 \Delta\rho(|\phi|^2) \right] \phi = 0, \quad (3)$$

where $k = 2\pi/\lambda$ is the wavenumber of the laser radiation. It is assumed in SSPARAMA that at $z = 0$ the pulse train has a spherical phase front and a truncated intensity profile. For example, when truncated Gaussian pulses are propagated

$$\begin{aligned} \phi(x, y, 0) &= N_R \phi_R(x, y), & x^2 + y^2 &\leq 2a^2, \\ &= 0, & x^2 + y^2 &> 2a^2, \end{aligned} \quad (4)$$

where

$$\phi_R(x, y) = \frac{1}{a\sqrt{\pi}} e^{-[1 + (ika^2/f)] \{(x^2 + y^2)/a^2\}/2} \quad (5)$$

and N_k is a normalization constant insuring that Eq. (2) is satisfied at $z = 0$. Two scale lengths, a and f , are defined in Eq. (5). The scale length f , the initial curvature of the phase front, defines the distance from the aperture to the focal plane. At a distance a from the aperture center the beam intensity falls to $1/e$ of its maximum value, and the beam is truncated at $1/e^2$ of maximum intensity.

Altogether eight variable physical quantities, a , f , k , α , E_p , Δt_s , v_0 , and Ω appear in Eqs. (1) through (5). All variations will not however lead to a mathematically distinct problem. In SSPARAMA Eqs. (1) through (5) are scaled so that distinct propagation problems are defined in terms of five dimensionless parameters. The program is designed to accept either the set of data with dimensions or the dimensionless set, and both sets are printed out.

The scaling of Eqs. (1) through (5) is carried out via the coordinate transformations

$$\tilde{x} \equiv \frac{x}{a}, \quad \tilde{y} \equiv \frac{y}{a}, \quad \tilde{z} \equiv \frac{z}{f} \quad (6)$$

and the variable transformation

$$\tilde{\phi}(\tilde{x}, \tilde{y}, \tilde{z}) \equiv a \phi(x, y, z). \quad (7)$$

By multiplying Eq. (3) through by a^3 , one can write the propagation equation in a form which identifies the five dimensionless parameters characterizing propagation in SSPARAMA:

$$\left\{ 2iN_k \frac{\partial}{\partial \tilde{z}} + \frac{\partial^2}{\partial \tilde{x}^2} + \frac{\partial^2}{\partial \tilde{y}^2} - N_k N_c e^{-N_\alpha \tilde{z}} \sum_{n=1}^{\infty} \left| \tilde{\phi} \left[\tilde{x} - \frac{2n}{N_o} (1 + N_s \tilde{z}), \tilde{y}, \tilde{z} \right] \right|^2 \right\} \tilde{\phi} = 0. \quad (8)$$

The five parameters, N_k , N_c , N_α , N_o , and N_s , are defined as

$$N_k = ka^2/f, \quad (9)$$

$$N_c = \frac{3Nk(\gamma - 1)\alpha/E_p}{c_s^2 a^2}, \quad (10)$$

$$N_\alpha = \alpha f, \quad (11)$$

$$N_o = \frac{2a}{v_0 \Delta t_s}, \quad (12)$$

and

$$N_s = \Omega f/v_0. \quad (13)$$

N_k is the Fresnel number of the free-propagation problem, and N_c , N_α , N_o , and N_s are coupling strength, absorption, overlap, and sluing parameters respectively. N_o was introduced by Wallace and Lilly [4] and called the pulses-per-flow-time parameter. It measures the number of preceding pulses which have heated the air across the beam

aperture as the pulse under study begins to propagate. The solution to Eq. (8) is obtained subject to the energy normalization

$$\int |\tilde{\phi}(\tilde{x}, \tilde{y}, 0)|^2 d\tilde{x} d\tilde{y} = 1 \quad (14)$$

and the initial condition

$$\tilde{\phi}(\tilde{x}, \tilde{y}, 0) = |\tilde{\phi}| e^{-iN_h(\tilde{x}^2 + \tilde{y}^2)/2}, \quad (15)$$

where $|\tilde{\phi}| = 0$ for $\tilde{x}^2 + \tilde{y}^2 > 2$.

Equations (8), (14), and (15) are numerically solved in SSPARAMA on a 64-by-64 grid in the $\tilde{x}\tilde{y}$ plane. Since one would like to use as much of the computational grid as possible to describe the variations in beam intensity, a scheme for adapting the coordinate grid to the propagation must be used. For example, as the beam propagates, the initial focusing causes the beam intensity pattern to decrease in size until the negative lensing effects of the heated atmosphere accumulate to thermally defocus it. Moreover, since the wind removes heated air from the path of the beam from left to right, a thermal gradient is established that deflects the beam from right to left. If the computational grid were not moved or changed in size as the beam intensity was calculated from aperture to focal plane, the intensity pattern would either be poorly sampled as it decreased in size or it would expand or deflect to reach the boundary of the grid and invalidate the calculation.

A technique for adapting the computational grid to local changes in the size or location of the beam intensity pattern has been developed by Herrmann and Bradley [5]. A slightly modified form of their technique has been incorporated into SSPARAMA and will be described in the next section of this report. In the third section the numerical procedures used in SSPARAMA will be described, and in the fourth section the code usage will be explained.

COORDINATE-SYSTEM ADAPTION

The dimensionless form of the propagation equation can be rewritten more compactly as

$$[2iN_h \partial_{\tilde{z}} + \partial_{\tilde{x}}^2 + \partial_{\tilde{y}}^2 + k^2 a^2 (n^2 - 1)] \tilde{\phi} = 0, \quad (16)$$

where $n^2 - 1$, the nonlinear index of refraction, depends on $\tilde{\phi}$ as given by Eq. (8). The $\tilde{x}\tilde{y}\tilde{z}$ coordinate system is normalized to the constant lengths a and f , and is fixed in space. In this system therefore the beam will lie symmetrically about the origin of the $\tilde{x}\tilde{y}$ plane only at $\tilde{z} = 0$ with an extent of order 1 (see, for example, Eq. (15)). When $\tilde{z} \neq 0$, a new set of xy coordinates is needed to maintain the two properties that the beam be centered about the xy coordinate origin and be of order 1 in extent. In general, one can relate the xy and $\tilde{x}\tilde{y}$ coordinates by a set of scale parameters D_1 and D_2 and a deflection parameter X , which are functions of $\tilde{z}$. Since one would like to solve Eq. (16) in a set of coordinates that adapt to changes in beam size and direction, the coordinate transformation

must be related to these beam changes as determined by the linear and quadratic terms of the phase front. By analogy therefore with the transformation to dimensionless parameters, one must perform simultaneous coordinate and variable transformations. The form of these transformations is suggested by linear propagation theory:

$$x = \frac{\tilde{x} - X}{\sqrt{D_1}}, \quad (17)$$

$$y = \frac{\tilde{y}}{\sqrt{D_2}}, \quad (18)$$

$$z = \frac{\tilde{z}}{N_k}, \quad (19)$$

and

$$\tilde{\phi} = \frac{\psi}{\sqrt[4]{D_1 D_2}} e^{i(\tilde{\alpha}_1 \tilde{x}^2 + \tilde{\alpha}_2 \tilde{y}^2 + \tilde{\beta} \tilde{x} + \tilde{\gamma}_1 + \tilde{\gamma}_2)}. \quad (20)$$

The constant scale change from $\tilde{z}$ to z is done for convenience to eliminate N_k from the z -derivative term in Eq. (16):

$$2iN_k \partial_{\tilde{z}} \rightarrow 2i \partial_z.$$

The factor $1/\sqrt[4]{D_1 D_2}$ is removed from $\tilde{\phi}$ to insure the form invariance of the energy normalization:

$$\int |\tilde{\phi}|^2 d\tilde{x} d\tilde{y} = \int |\psi|^2 dx dy = 1. \quad (21)$$

When Eqs. (17) through (20) are substituted into Eq. (16) and when the nonlinear term is of negligible size and the beam has a Gaussian profile, D_1 , D_2 , X , $\tilde{\alpha}_1$, $\tilde{\alpha}_2$, $\tilde{\beta}$, $\tilde{\gamma}_1$, and $\tilde{\gamma}_2$ as functions of z can be analytically determined for all z . However, when the nonlinear term is important or when a non-Gaussian beam is propagated, the $\tilde{\alpha}$'s and $\tilde{\beta}$, which represent the effective quadratic and linear phase changes throughout the xy plane, can no longer be so determined. One must adopt a more limited strategy for the employment of Eqs. (17) through (20).

Consider, for example, that the quantities D_1 , D_2 , X , $\tilde{\alpha}_1$, $\tilde{\alpha}_2$, $\tilde{\beta}$, $\tilde{\gamma}_1$, and $\tilde{\gamma}_2$ are known at $z = z_0$ and that their dependence on z is to be analytically determined as one propagates to a neighboring xy plane at $z_0 + \Delta z$. Since

$$\partial_{\tilde{x}}^2 = \frac{1}{D_1} \partial_x^2, \quad (22)$$

$$\partial_{\tilde{y}}^2 = \frac{1}{D_2} \partial_y^2, \quad (23)$$

and

$$\partial_z = \frac{1}{N_k} \left[\partial_z - \left(\frac{x}{2} \partial_z \ln D_1 + \frac{\partial_z X}{\sqrt{D_1}} \right) \partial_x - \frac{y}{2} \partial_z \ln D_2 \partial_y \right], \quad (24)$$

one finds that

$$\begin{aligned} & [2iN_k \partial_z + \partial_x^2 + \partial_y^2 + k^2 a^2 (n^2 - 1)] \frac{\psi}{\sqrt{D_1 D_2}} e^{i(\tilde{\alpha}_1 \tilde{x}^2 + \tilde{\alpha}_2 \tilde{y}^2 + \tilde{\beta} \tilde{x} + \tilde{\gamma}_1 + \tilde{\gamma}_2)} \\ &= \frac{e^{i(\tilde{\alpha}_1 \tilde{x}^2 + \tilde{\alpha}_2 \tilde{y}^2 + \tilde{\beta} \tilde{x} + \tilde{\gamma}_1 + \tilde{\gamma}_2)}}{\sqrt{D_1 D_2}} \left\{ 2i \left(\partial_z - \frac{x}{2} \partial_z \ln D_1 \partial_x - \frac{1}{\sqrt{D_1}} \partial_z X \partial_x - \frac{y}{2} \partial_z \ln D_2 \partial_y \right) \right. \\ &\quad - \frac{1}{2} (\partial_z \ln D_1 + \partial_z \ln D_2) - 2\partial_z (\tilde{\gamma}_1 + \tilde{\gamma}_2) + \frac{1}{D_1} \partial_x^2 - [2\tilde{\alpha}_1 (\sqrt{D_1} x + X) + \tilde{\beta}]^2 \\ &\quad + \frac{2i}{\sqrt{D_1}} [2\tilde{\alpha}_1 (\sqrt{D_1} x + X) + \tilde{\beta}] \partial_x + 2i\tilde{\alpha}_1 + \frac{1}{D_2} \partial_y^2 - 4\tilde{\alpha}_2^2 D_2 y^2 \\ &\quad \left. + 4i\tilde{\alpha}_2 y \partial_y + 2i\tilde{\alpha}_2 + k^2 a^2 (n^2 - 1) \right\} \psi = 0. \quad (25) \end{aligned}$$

For vanishingly small $n^2 - 1$ and for a real Gaussian profile $\psi(x, y, z_0)$ one would determine D_1 , D_2 , X , $\tilde{\alpha}_1$, $\tilde{\alpha}_2$, $\tilde{\beta}$, $\tilde{\gamma}_1$, and $\tilde{\gamma}_2$ from the requirement that Eq. (25) be capable of being put in the form

$$\left[2i \partial_z + \frac{1}{D_1} (\partial_x^2 + 1 - x^2) + \frac{1}{D_2} (\partial_y^2 + 1 - y^2) + k^2 a^2 (n^2 - 1) \right] \psi = 0. \quad (26)$$

Then, as ψ was propagated to $z_0 + \Delta z$, it would acquire no z dependence and would remain real and Gaussian; that is, all of the z dependence of ϕ would have been accounted for in $D_1, \dots, \tilde{\gamma}_2$.

For the imaginary terms of Eq. (25) other than $2i\partial_z$ to vanish, the quantities D_1 , D_2 , and X , which determine the scale and location of the xyz coordinate system, must satisfy the equations

$$\partial_z \ln D_1 = 4\tilde{\alpha}_1, \quad (27)$$

$$\partial_z \ln D_2 = 4\tilde{\alpha}_2, \quad (28)$$

and

$$\partial_z X = 2\tilde{\alpha}_1 X + \tilde{\beta}. \quad (29)$$

On the other hand, for the real terms involving ∂_x and ∂_y to vanish and for the scale functions D_1 and D_2 to be factorable from the remaining x and y terms respectively, the phase functions $\tilde{\alpha}_1$, $\tilde{\alpha}_2$, $\tilde{\beta}$, $\tilde{\gamma}_1$, and $\tilde{\gamma}_2$ must satisfy the set of equations

$$2D_1 \partial_z \tilde{\alpha}_1 + 4\tilde{\alpha}_1^2 D_1 = \frac{1}{D_1}, \quad (30)$$

$$2D_2 \partial_z \tilde{\alpha}_2 + 4\tilde{\alpha}_2^2 D_2 = \frac{1}{D_2}, \quad (31)$$

$$\partial_z \tilde{\beta} + 2X \partial_z \tilde{\alpha}_1 + 2\tilde{\alpha}_1 (2\tilde{\alpha}_1 X + \tilde{\beta}) = 0, \quad (32)$$

$$2\partial_z \tilde{\gamma}_1 + 2X^2 \partial_z \tilde{\alpha}_1 + 2X \partial_z \tilde{\beta} + (2\tilde{\alpha}_1 X + \tilde{\beta})^2 = -\frac{1}{D_1}, \quad (33)$$

and

$$2\partial_z \tilde{\gamma}_2 = -\frac{1}{D_2}. \quad (34)$$

Thus Eqs. (27) through (34) will determine all of the z dependence of $\tilde{\phi}$ when $\psi(x, y, z_0)$ is real and a Gaussian function of x and y and there is no lensing effect caused by heating of the atmosphere; that is, Eqs. (27) through (34) will describe beam focusing in the absence of diffraction and nonlinear media phenomena. They are of more limited utility when such phenomena are present. In this case, during the displacement of ϕ from z_0 to $z_0 + \Delta z$, linear and quadratic phase changes will arise from two sources. As a result of focusing at $z = z_0$, the initial phases $\tilde{\alpha}_1(z_0)$, $\tilde{\alpha}_2(z_0)$, and $\tilde{\beta}(z_0)$ will become $\tilde{\alpha}_1(z_0 + \Delta z)$, $\tilde{\alpha}_2(z_0 + \Delta z)$, and $\tilde{\beta}(z_0 + \Delta z)$ through the solution to Eqs. (27) through (34). In addition however ψ at $z_0 + \Delta z$ will acquire linear and quadratic phases, $\Delta\tilde{\beta}$, $\Delta\tilde{\alpha}_1$, and $\Delta\tilde{\alpha}_2$ respectively, as a result of diffraction and thermal blooming. Thus at $z_0 + \Delta z$ a new factorization of $\tilde{\phi}$ must be made, namely,

$$\tilde{\phi}(\tilde{x}, \tilde{y}, \tilde{z}_0 + \Delta\tilde{z}) \equiv \frac{\psi'(x, y, z_0 + \Delta z)}{\sqrt[4]{D_1(z_0 + \Delta z)D_2(z_0 + \Delta z)}} e^{i[\tilde{\alpha}_1'(z_0 + \Delta z)\tilde{x}^2 + \tilde{\alpha}_2'(z_0 + \Delta z)\tilde{y}^2 + \tilde{\beta}'(z_0 + \Delta z)\tilde{x} + \tilde{\gamma}_1' + \tilde{\gamma}_2']}, \quad (35)$$

if ψ' , which is to be propagated from $z_0 + \Delta z$ to $z_0 + \Delta z + \Delta z'$, is not to initially have quadratic or linear phase terms. After each step in propagation therefore $\tilde{\alpha}_1$, $\tilde{\alpha}_2$, and $\tilde{\beta}$ must be redefined as

$$\tilde{\alpha}_1'(z_0 + \Delta z) = \tilde{\alpha}_1(z_0 + \Delta z) + \Delta\tilde{\alpha}_1, \quad (36)$$

$$\tilde{\alpha}_2'(z_0 + \Delta z) = \tilde{\alpha}_2(z_0 + \Delta z) + \Delta\tilde{\alpha}_2, \quad (37)$$

and

$$\tilde{\beta}'(z_0 + \Delta z) = \tilde{\beta}(z_0 + \Delta z) + \Delta\tilde{\beta} \quad (38)$$

In order to adapt the coordinate-system determination from Eqs. (27) through (29) to changes in phase that result from focusing, diffraction, and thermal blooming.

In SSPARAMA, ψ is propagated from one z plane to another by finite-differencing a phase-transformed version of Eq. (26). Then $\Delta\alpha_1$, $\Delta\alpha_2$, and $\Delta\beta$ are found in the xyz coordinate system using the method of phase minimization discussed by Herrmann and Bradley [5]. One requires that

$$\int_{z=z_0+\Delta z} |\psi|^2 [\nabla(\Delta\alpha_1 x^2 + \Delta\alpha_2 y^2 + \Delta\beta x - \gamma)]^2 dx dy = \text{minimum}, \quad (39)$$

where $\psi(x, y, z_0 + \Delta z) \equiv |\psi|e^{i\gamma}$. It follows that

$$\Delta\alpha_1 = \frac{D_1 E - B_1 C_1}{2(A_1 E - B_1^2)}, \quad (40)$$

$$\Delta\beta = \frac{A_1 C_1 - B_1 D_1}{A_1 E - B_1^2}, \quad (41)$$

and

$$\Delta\alpha_2 = \frac{D_2}{2A_2}, \quad (42)$$

where

$$A_1 \equiv \int x^2 |\psi|^2 dx dy, \quad A_2 \equiv \int y^2 |\psi|^2 dx dy, \quad (43)$$

$$B_1 \equiv \int x |\psi|^2 dx dy, \quad (44)$$

$$C_1 \equiv \text{Im} \int \psi^* \partial_x \psi dx dy, \quad (45)$$

$$D_1 \equiv \text{Im} \int x \psi^* \partial_x \psi dx dy, \quad D_2 \equiv \text{Im} \int y \psi^* \partial_y \psi dx dy, \quad (46)$$

and

$$E \equiv \int |\psi|^2 dx dy = 1. \quad (47)$$

The factorization

$$\psi(x, y, z_0 + \Delta z) \equiv \psi' e^{i(\Delta\alpha_1 x^2 + \Delta\alpha_2 y^2 + \Delta\beta x)} \quad (48)$$

will then define ψ' at $z_0 + \Delta z$ as a wave function of minimum quadratic and linear phase. In particular, if ψ is exactly a Gaussian beam, ψ' will be real.

The relationship between $\{\Delta\alpha_1, \Delta\alpha_2, \Delta\beta\}$ and $\{\Delta\tilde{\alpha}_1, \Delta\tilde{\alpha}_2, \Delta\tilde{\beta}\}$ is found by substituting Eqs. (17) and (18) into Eq. (48):

$$\Delta\tilde{\alpha}_1 = \frac{\Delta\alpha_1}{D_1}, \quad (49)$$

$$\Delta\tilde{\alpha}_2 = \frac{\Delta\alpha_2}{D_2}, \quad (50)$$

and

$$\Delta\tilde{\beta} = \frac{\Delta\beta}{\sqrt{D_1}} - \frac{2\Delta\alpha_1 X}{D_1}. \quad (51)$$

A similar set of equations will hold between $\{\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\beta}\}$ and $\{\alpha_1, \alpha_2, \beta\}$, which are computed directly in the xyz coordinate system. When reexpressed in terms of α_1, α_2 and β , Eqs. (27) through (29) become

$$\partial_z D_1 = 4\alpha_1, \quad (52)$$

$$\partial_z D_2 = 4\alpha_2, \quad (53)$$

and

$$\partial_z X = \frac{\beta}{\sqrt{D_1}}, \quad (54)$$

and Eqs. (30) through (32) transform into

$$\partial_z \alpha_1 = \frac{1}{2D_1} (1 + 4\alpha_1^2), \quad (55)$$

$$\partial_z \alpha_2 = \frac{1}{2D_2} (1 + 4\alpha_2^2), \quad (56)$$

and

$$\partial_z \beta = \frac{2\alpha_1 \beta}{D_1}. \quad (57)$$

Eqs. (52) through (57) must be solved in terms of initial values at z_0 . The solutions are

$$D_{1,2}(z) = D_{1,2}(z_0) \left\{ \left[1 + \frac{2\alpha_{1,2}(z_0)}{D_{1,2}(z_0)} (z - z_0) \right]^2 + \left[\frac{z - z_0}{D_{1,2}(z_0)} \right]^2 \right\}, \quad (58)$$

$$\alpha_{1,2}(z) = \alpha_{1,2}(z_0) + \frac{1}{2} \{1 + [2\alpha_{1,2}(z_0)]^2\} \frac{z - z_0}{D_{1,2}(z_0)}, \quad (59)$$

$$\beta(z) = \beta(z_0) \sqrt{\left[1 + \frac{2\alpha_1(z_0)}{D_1(z_0)}(z - z_0)\right]^2 + \left[\frac{z - z_0}{D_1(z_0)}\right]^2}, \quad (60)$$

and

$$X(z) = X(z_0) + \frac{\beta(z_0)}{\sqrt{D_1(z_0)}}(z - z_0). \quad (61)$$

Finally the procedure for solving Eq. (26) in SSPARAMA is similar to the one described in an earlier report [2]. A phase transformation on ψ is made:

$$\Phi(x, y, z) \equiv \psi(x, y, z) e^{-(i/2) \int_{z_0}^z g(x, y, z') dz'}, \quad (62)$$

where

$$g(x, y, z) \equiv \frac{1}{D_1}(1 - x^2) + \frac{1}{D_2}(1 - y^2) + k^2 a^2 (n^2 - 1). \quad (63)$$

The equation for Φ follows from Eq. (26):

$$[2i\partial_z + H(x, y, z)]\Phi = 0, \quad (64)$$

where

$$H = e^{-(i/2) \int_{z_0}^z g dz'} \left(\frac{1}{D_1} \partial_x^2 + \frac{1}{D_2} \partial_y^2 \right) e^{(i/2) \int_{z_0}^z g dz'}. \quad (65)$$

By picking z'_0 to lie between z_0 and $z_0 + \Delta z$, one can propagate Φ from z_0 to $z_0 + \Delta z$, with first-order accuracy, by solving the equation

$$[2i\partial_z + H(x, y, z'_0)]\Phi = \left(2i\partial_z + \frac{1}{D_1} \partial_x^2 + \frac{1}{D_2} \partial_y^2 \right) \Phi = 0. \quad (66)$$

Equation (66) is solved by Fourier transforming Φ [6],

$$\tilde{\Phi}(k_1, k_2, z_0) \equiv \int e^{i(k_1 x + k_2 y)} \Phi(x, y, z_0) dx dy, \quad (67)$$

and propagating $\tilde{\Phi}$ to $z_0 + \Delta z$:

$$\tilde{\Phi}(k_1, k_2, z_0 + \Delta z) = \tilde{\Phi}(k_1, k_2, z_0) e^{(i/2) \{ k_1^2 \int_{z_0}^{z_0 + \Delta z} [1/D_1(z)] dz + k_2^2 \int_{z_0}^{z_0 + \Delta z} [1/D_2(z)] dz \}}. \quad (68)$$

The inverse transformation to Eq. (67) then yields Φ , and Eq. (62) yields $\psi(x, y, z_0 + \Delta z)$.

NUMERICAL PROCEDURES

The phase function $g(x, y, z)$ of Eq. (63) can be written more usefully in the form

$$g = \frac{g_1(x)}{D_1(z)} + \frac{g_2(y)}{D_2(z)} - \frac{g_3(x, y, z)}{\sqrt{D_1(z)D_2(z)}}, \quad (69)$$

where

$$g_1(x) \equiv 1 - x^2, \quad (70)$$

$$g_2(y) \equiv 1 - y^2, \quad (71)$$

and

$$g_3(x, y, z) \equiv N_k N_c e^{-N_a N_k z} \sum_{n=1}^{\infty} \left| \Phi \left[x - \frac{2n}{N_0 \sqrt{D_1(z)}} (1 + N_s N_k z), y, z \right] \right|^2. \quad (72)$$

This expression for g_3 is found by substituting the new variables x, y, z , and Φ into Eq. (8). The phase integral

$$\Delta\theta \equiv \int_{z_0}^z g(x, y, z') dz'$$

appearing in Eq. (62) can now be partially evaluated and expressed in the form

$$\Delta\theta = g_1(x)\Delta Z_1 + g_2(y)\Delta Z_2 - \int_{z_0}^z \frac{g_3(x, y, z)}{\sqrt{D_1(z)D_2(z)}} dz, \quad (73)$$

where

$$\Delta Z_{1,2} \equiv \int_{z_0}^z \frac{dz'}{D_{1,2}(z')} = \tanh^{-1} \left(\left\{ 1 + [2\alpha_{1,2}(z_0)]^2 \right\} \frac{z' - z_0}{D_{1,2}(z_0)} + 2\alpha_{1,2}(z_0) \right) \Big|_{z'=z_0}^{z'=z}. \quad (74)$$

The differential quantities ΔZ_1 and ΔZ_2 are similarly named as the coordinate differential ΔZ that was used in earlier code calculations which involved only a single scaling function $D(z)$.

To complete the evaluation of $\Delta\theta$, one must know the z dependence of g_3 , that is, the z dependence of $|\Phi|^2$. Two options are provided in SSPARAMA, for evaluating $\Delta\theta$, depending on whether one has determined $|\Phi|^2$ at one or both of the integration

endpoints. The procedures work as follows: Suppose first that the solution for $\psi(x, y, z_0)$ has been obtained. Then one can compute $g(x, y, z_0)$, since $|\Phi(x, y, z_0)|^2 = |\psi(x, y, z_0)|^2$. To find $\Phi(x, y, z_0)$, however, one must evaluate

$$\Delta\theta' \equiv \int_{z_0}^{z'_0} g(x, y, z') dz', \quad (75)$$

where z'_0 lies between z_0 and the plane $z_0 + \Delta z$ to which one would like to propagate ψ . If ψ is known only at z_0 , the zeroth-order approximation

$$\Delta\theta' \approx g_1(x)\Delta Z'_1 + g_2(y)\Delta Z'_2 - g_3(x, y, z_0)\Delta Z'_{12} \quad (76)$$

must be made, where

$$\Delta Z'_{12} \equiv \int_{z_0}^{z'_0} \frac{dz'}{\sqrt{D_1(z')D_2(z')}}. \quad (77)$$

Equation (66) can now be solved for $\Phi(x, y, z_0 + \Delta z)$ by the use of Fourier transformations. Finally on performance of the phase integral

$$\Delta\theta'' \equiv \int_{z'_0}^{z_0 + \Delta z} g(x, y, z') dz' \quad (78)$$

$\psi(x, y, z_0 + \Delta z)$ can be obtained from $\Phi(x, y, z_0 + \Delta z)$. In keeping with the accuracy with which $\Delta\theta'$ was approximated, $\Delta\theta''$ can be approximately evaluated as

$$\Delta\theta'' \approx g_1(x)\Delta Z''_1 + g_2(y)\Delta Z''_2 - g_3(x, y, z_0 + \Delta z)\Delta Z''_{12}. \quad (79)$$

The differentials $\Delta Z''_1$, $\Delta Z''_2$, and $\Delta Z''_{12}$ are defined by the integrals of Eqs. (74) and (77) with the integration limits as specified in Eq. (78).

Suppose however that initially both $\psi(x, y, z_0)$ and $\psi(x, y, z'_0)$ are known and that the values of ψ at z_0 are to be propagated to the plane at $z_0 + \Delta z$. In this case the phase integrals defined in Eqs. (75) and (78) can be approximated using the integration formula

$$\int_{x_0}^{x_0 + \Delta x} f(x)g(x) dx \approx w_1 f(x_0) + w_2 f(x_0 + \Delta x), \quad (80)$$

which has first-order instead of zeroth-order accuracy. The weights w_1 and w_2 are thus determined such that equality will hold in Eq. (80) whenever f is a linear function of x :

$$w_1 = \left(1 + \frac{2x_0}{\Delta x}\right) \int_{x_0}^{x_0 + \Delta x} g(x) dx - \frac{2}{\Delta x} \int_{x_0}^{x_0 + \Delta x} xg(x) dx \quad (81)$$

and

$$\omega_2 = \frac{2}{\Delta x} \int_{x_0}^{x_0 + \Delta x} x g(x) dx - \frac{2x_0}{\Delta x} \int_{x_0}^{x_0 + \Delta x} g(x) dx. \quad (82)$$

Then, for example, in place of Eq. (76) one would have that

$$\Delta\theta' \approx g_1(x)\Delta Z'_1 + g_2(y)\Delta Z'_2 - g_3(x, y, z_0)\Delta Z'_3 - g_3(x, y, z'_0)\Delta Z'_4, \quad (83)$$

where $\Delta Z'_3$ and $\Delta Z'_4$ are related through Eqs. (81) and (82) to $\Delta Z'_{12}$ and an integration over the function $z/\sqrt{D_1(z)D_2(z)}$:

$$\Delta Z'_3 = \frac{z'_0 + z_0}{z'_0 - z_0} \Delta Z'_{12} - \frac{2}{z'_0 - z_0} \int_{z_0}^{z'_0} \frac{z' dz'}{\sqrt{D_1(z')D_2(z')}} \quad (84)$$

and

$$\Delta Z'_4 = \frac{2}{z'_0 - z_0} \left[\int_{z_0}^{z'_0} \frac{z' dz'}{\sqrt{D_1(z')D_2(z')}} - z_0 \Delta Z'_{12} \right]. \quad (85)$$

Although integrations over D_1^{-1} and D_2^{-1} can be carried out analytically in terms of inverse hyperbolic tangents (as in Eq. (74)), integrals over $1/\sqrt{D_1 D_2}$ produce elliptic functions. Both sets of integrations are handled in SSPARAMA numerically, with third-order accuracy, using a second integration formula:

$$\int_{x_0}^{x_0 + \Delta x} f(x) dx \approx \frac{\Delta x}{2} [f(x_0 + \Delta x_1) + f(x_0 + \Delta x_2)], \quad (86)$$

where $\Delta x_1 \equiv (1 - 1/\sqrt{3})\Delta x/2$ and $\Delta x_2 \equiv (1 + 1/\sqrt{3})\Delta x/2$. Again, as an example, consider Eqs. (84) and (85) and define

$$f_1 \equiv \frac{1}{\sqrt{D_1(z_1)D_2(z_1)}} \quad (87)$$

and

$$f_2 \equiv \frac{1}{\sqrt{D_1(z_2)D_2(z_2)}}, \quad (88)$$

where $z_1 \equiv z_0 + (1 - 1/\sqrt{3})(z'_0 - z_0)/2$ and $z_2 \equiv z_0 + (1 + 1/\sqrt{3})(z'_0 - z_0)/2$. One can complete the numerical evaluation of $\Delta Z'_3$ and $\Delta Z'_4$ by rewriting Eqs. (84) and (85) with the use of Eq. (86), in terms of f_1 and f_2 :

$$\Delta Z'_3 = \frac{z'_0 - z_0}{2\sqrt{3}} (f_1 - f_2) \quad (89)$$

and

$$\begin{aligned} \Delta Z'_4 &= \frac{z'_0 - z_0}{2} \left[\left(1 - \frac{1}{\sqrt{3}}\right) f_1 + \left(1 + \frac{1}{\sqrt{3}}\right) f_2 \right] \\ &= (z'_0 - z_0) \left(\frac{f_1 + f_2}{2} \right) - \Delta Z'_3. \end{aligned} \quad (90)$$

The procedure by which Eqs. (80) through (90) are employed requires that two sets of values of ψ be stored at any time by SSPARAMA. At the beginning of the propagation step described above, the two arrays contain the values of $\psi(x, y, z_0)$ and $\psi(x, y, z'_0)$, where $z_0 < z'_0 < z_0 + \Delta z$. At the end of the propagation step the values of $\psi(x, y, z_0)$ have been replaced by $\psi(x, y, z_0 + \Delta z)$. These new values can then be used to propagate $\psi(x, y, z'_0)$ to $\psi(x, y, z'_0 + \Delta z')$, where now $z'_0 < z_0 + \Delta z < z'_0 + \Delta z'$. The process of alternatively propagating one and then the other of the two arrays is repeated until the focal plane, defined by the initial beam curvature, is reached.

Since both arrays are initially assigned the values $\psi(x, y, 0)$, the process of propagating one array past the other cannot begin until after the first propagation step. The first z step is therefore taken using Eqs. (76) and (79) to determine $\Delta\theta'$ and $\Delta\theta''$. In general the incremental steps Δz are selected in SSPARAMA according to a criterion that the phase changes induced by g_3 as computed from Eq. (76) be no larger than some pre-assigned value of order 1 for all x and y . However, to carry out the first advancement of ψ at $z_0 = 0$, half of the initially computed Δz value is used. This leapfrog procedure is summarized for the first few z steps in Fig. 1.

The advantage conveyed by using Eqs. (76) and (79) to evaluate the phase integrals $\Delta\theta'$ and $\Delta\theta''$ is that only one ψ array is needed in carrying out the calculation. Because of the reduced accuracy in computing $\Delta\theta'$ and $\Delta\theta''$, however, smaller z steps are in principle required to obtain the same results as when two arrays at different z planes are used. To allow a quantitative comparison of these two procedures, both options for propagating ψ were installed in SSPARAMA and can be selected according to the value of one of the input parameters to the code. For the same reason, another input parameter is also available that allows one to adapt or not adapt the coordinate system to the amount of diffraction or thermal blooming occurring during beam propagation.

PROGRAM OPERATION

This section will describe the input parameters required to run SSPARAMA and explain the data included in the output. A complete listing of SSPARAMA is included in Appendix A.

To use program SSPARAMA, two input cards are required. The first specifies certain numerical parameters and selects various program options, and the second defines the

NRL REPORT 8074

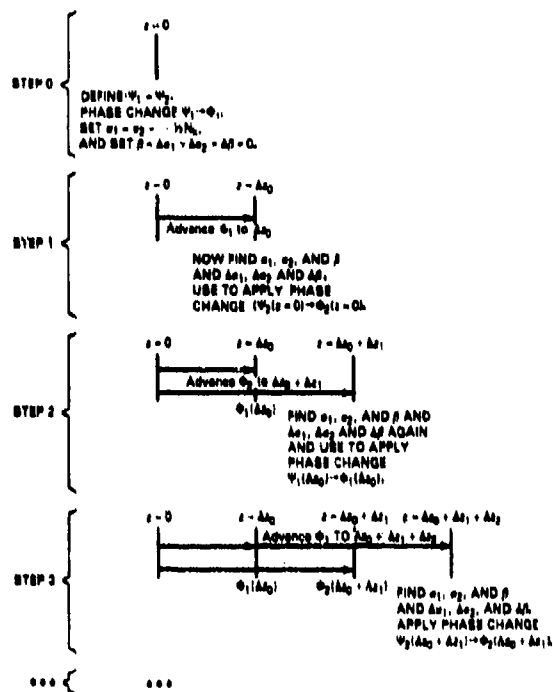


Fig. 1—Leapfrog procedure for advancing the wave function Φ

particular physical situation. This second card can contain the actual physical parameters or a set of dimensionless parameters.

First Input Card

The parameters read from the first card are listed in Table 1. A description of each of these parameters is as follows:

Table 1—Parameters Specified by the First Input Card

Columns	Name	Format	Columns	Name	Format
1-5	PHIMXX	F5.0	36-40	NPM	I5
6-10	ROCLT	F5.0	41-45	NBM	I5
11-15	HXY	F5.0	46-50	NPLOT	I5
16-20	NXY	I5	51-55	NCT	I5
21-25	NCW	I5	56-60	NRS	I5
26-30	NAD	I5	61-65	NPUNCH	I5
31-35	NMS	I5	76-80	NID	A6

WHITNEY, MADER, AND ULRICH

PHIMXX. This is the maximum allowed phase change in radians for any point in the computational grid at each z step. It is used to define the newly computed z increments HZN at each step, where

$$\frac{g_3(x, y, z)_{\max}}{\sqrt{D_1 D_2}} \text{HZN} = \text{PHIMXX},$$

in which $g_3(x, y, z)_{\max}$ is the maximum value in the computational grid of g_3 , given by Eq. (72). PHIMXX is nominally entered as 1.0. If more z steps are required, PHIMXX can be decreased. In this case the z increment is tied to the amount of heating in the atmosphere, becoming smaller automatically as large density changes take place or becoming large and efficient when near-vacuumlike propagation occurs. If HZN exceeds 0.1 of the total propagation distance, the smaller of these two z increments is used. If HZN at any time is less than 10^{-7} times the distance to be propagated, the program exits and an error message will be printed.

ROCULT. This is used when propagating uniform circular beamshapes with an obscuring disk or a uniform rectangular beamshape. In the former case ROCULT is the ratio of the occulting radius to the total radius. For a rectangle, it is the ratio of the y to the x dimension. ROCULT is used only when NBM equals 4 or 5.

HXY. This parameter defines the size of the computational grid relative to the aperture radius by

$$\Delta x = \Delta y = \text{HXY}$$

where Δx and Δy are the sizes of individual computational cells, which start out square. Depending on the beamshape, values between 0.1 and 0.3 are typical.

NXY. This is the number of individual computational cells along the edge of the entire computational grid. The FFT routine is more efficient when NXY is a power of 2, and NXY is normally entered as 64.

NCW. This parameter permits CW propagation to be included by allowing the summation in Eq. (72) to be replaced by an integral [7]. Before the summation is replaced, Eq. (72) can be written in terms of physical parameters as

$$\frac{3N(\gamma - 1)k^2 \alpha E_p e^{-\alpha z}}{c_s^2} \sum_{n=1}^{\infty} |\Phi[x - n(v_0 + \Omega z)\Delta t, y, z]|^2.$$

This summation is performed when NCW = 0. When NCW = 1, the program is in the CW mode, and Eq. (72) is replaced by

$$\frac{3N(\gamma - 1)k^2 \alpha P e^{-\alpha z} \sqrt{D_1}}{c_s^2 (v_0 + \Omega z)} \int_{-\infty}^0 |\Phi(x + x', y, z)|^2 dx',$$

where P is the average power of a CW laser ($P = E_p/\Delta t$). The integration is performed using a simple trapezoid rule.

NAD. When NAD = 0, the coordinate system adaption is not included. When NAD = 1, it is included.

NMS. When NMS = 0, the midplane integrations are not used. When NMS = 1, they are used.

NPM. When NPM = -1, the second data card contains physical parameters. When NPM = +1, the second card contains dimensionless parameters.

NBM. This parameter selects one of the five beamshapes available within the program:

NBM = 0 -- Infinite Gaussian, with WIDTH (a parameter read from the second input card) being the e^{-1} intensity radius;

NBM = 1 -- Truncated Gaussian, with WIDTH being the e^{-1} intensity radius, truncated at $\sqrt{2} \times \text{WIDTH}$ or e^{-2} intensity radius;

NBM = 2 -- Uniform circular aperture, with WIDTH being the actual aperture radius;

NBM = 3 -- Uniform square aperture, with WIDTH being the dimension from the center of the square to the edge (half-side dimension) in the x or y direction;

NBM = 4 -- Uniform circular aperture and an occulting disk, with WIDTH being the total aperture radius and, as stated previously, with ROCULT being the ratio giving the occulting disk radius;

NBM = 5 -- Uniform rectangular aperture, with WIDTH being the half-side x dimension and ROCULT being the ratio giving the y dimension.

NPLOT. This determines the type and the number of plots given in the output:

NPLOT = 0 -- No plots;

NPLOT = 1 -- Final contour plot only;

NPLOT = 2 -- Final contour plot plus a plot of average intensity and peak intensity versus z ;

NPLOT = 3 -- Preceding plots plus a plot of flux and area versus irradiance;

NPLOT = 4 -- Preceding plots plus a contour plot of aperture intensity;

NPLOT = 5 -- Preceding plots plus Fourier-transform contour plots of aperture and final intensity distributions.

NCT. This determines the contour levels used in the contour plots:

NCT = 0 -- Contour plots use contour levels with 10% increments;

NCT = 1 -- Contour plots use 3-dB contours (0.5^n , $n = 1, 2, \dots, 10$).

NRS. When NRS = 1, the final contour plot is corrected and standardized according to an internal criterion, to remove the effects of different amounts of coordinate system

adaption in the x and y directions. When $NRS = 0$, this plot can appear with nonuniform axes.

NPUNCH. This determines whether there is a punched-card output:

NPUNCH = 0 — No punched-card output;

NPUNCH = 1 — Punched-card output for later data processing.

NID. Up to six characters can be used to identify a run or a series of runs on both the printed and punched output.

Second Input Card

The data contained on the second input card depend on the value of NPM . If $NPM = -1$, the physical parameters listed in Table 2 will be read. A description of each of these parameters is as follows:

OM. The slew rate in radians per second.

HT. The interval between pulses in seconds, or the reciprocal of the pulse repetition frequency (PRF). For CW propagation this should be set to 1 second.

ALPHA. The absorption coefficient α in km^{-1} .

ALPHAS. The scattering coefficient in km^{-1} . ALPHAS is used to compute the total extinction but is not included in the absorption that produces atmospheric heating.

WIDTH. The aperture radius a in centimeters. The particular definition is given in the preceding subsection for each value of NBM .

WN. The wavenumber $k = 2\pi/\lambda$ or $2\pi/\beta\lambda$, where β is the beam quality and λ is the beam wavelength in centimeters.

VO. The wind velocity v_0 in meters per second.

ENERGY. The individual pulse energy E_p in joules. For CW propagation ENERGY is the average power in watts.

F. The focal length in kilometers.

ZF. The distance at which the calculation is to be stopped in kilometers.

As already shown, the propagation is a function of five dimensionless parameters. Different combinations of the eight physical parameters, which are required to define

Table 2—Parameters Specified by the Second Input Card When $NPM = -1$

Columns	Name	Format
1-5	OM	F5.0
6-10	HT	F5.0
11-15	ALPHA	F5.0
16-20	ALPHAS	F5.0
21-30	WIDTH	E10.0
31-40	WN	E10.0
41-50	VO	E10.0
51-60	ENERGY	E10.0
61-70	F	E10.0
71-80	ZF	E10.0

these dimensionless parameters and which lead to the same values of the dimensionless parameters, will produce identical results. In order that a unique physical situation be specified, some physical quantities are also read from the second data card when NPM = +1 (Table 3). They are not used to define the physical situation but rather to assign units to the derived quantities at the end of the calculations. The quantities read when NPM = +1 are:

Table 3—Quantities Specified
by the Second Input Card
When NPM = +1

Columns	Name	Format
1-5	F	F5.0
6-10	HT	F5.0
11-20	PNA	E10.0
21-30	PNALF	E10.0
31-40	PNK	E10.0
41-50	PNO	E10.0
51-60	PNS	E10.0
61-70	PND	E10.0
71-80	PNZ	E10.0

F. Focal length in kilometers.

HT. Pulse interval Δt in seconds (=1 second for CW).

PNA. The f number = WIDTH/F.

PNALF. Absorption number, ALPHA/F.

PNK. Fresnel number, $WN \cdot \text{WIDTH}^2/F$.

PNO. Overlap number, $2\sqrt{2} \cdot \text{WIDTH}/(VO \cdot HT)$ for an infinite and truncated Gaussian beam and $2 \cdot \text{WIDTH}/(VO \cdot HT)$ for all other beam shapes.

PNS. Slew number, $OM \cdot F/VO$.

PND. Distortion number, $3Nk(\gamma - 1)\alpha/E_p/c_s^2 a v_0 \Delta t$.

PNZ. The ratio of the distance at which the calculation is to be stopped to the focal length, ZF/F .

Examples of Output

A series of multipulse runs was made varying the pulse spacing and energy so that the average power remained constant and using a number of average powers. The results of these runs are shown in Fig. 2 in the form of power optimization curves. The CW curve is included so that the convergence of the multipulse curves to the CW curve, as the limiting case when pulse interval is decreased, can be readily observed.

To test the SSPARAMA code in the CW mode, some comparison runs were made to check against some results obtained from Jan Herrmann of Lincoln Laboratory, who studied the propagation of a CW infinite Gaussian with a e^{-2} diameter of 70 cm. The absorption coefficient was 0.07 km^{-1} , with no scattering. The laser was twice-diffraction-limited DF with a wavenumber of $8.5 \times 10^3 \text{ cm}^{-1}$. Two cases were considered at focal lengths of 2, 5, and 10 km. The first case had a power of 10 MW, a wind speed of 250 m/s, and no slewing. The second case had 2 MW power, a 2-m/s wind, and a 0.02-s^{-1} slew. The results, consisting of the area containing 63% of the focal-plane power and of the peak intensity are summarized in Table 4. A_{rel} and I_{rel} compare these quantities with those that would have been obtained if there were no thermal blooming. The results for these highly bloomed cases agree within about 5% with those of Herrmann.

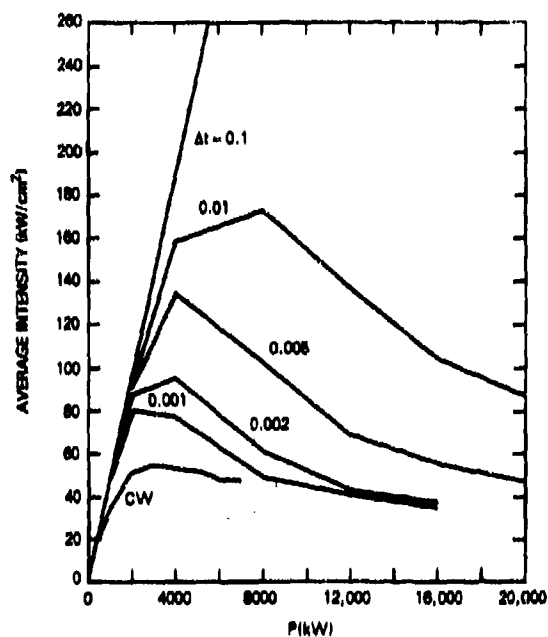


Fig. 2—SSPARAMA results ($F = 1$ km, diam = 70.7 cm ($1/e$), $\alpha = 0.1$ km $^{-1}$, $k = 2966$ cm $^{-1}$, $v_0 = 10$ m/s, and $\Omega = 0.1$)

Table 4—SSPARAMA Results for the Propagation of a CW Infinite Gaussian With a Wavenumber of 8500 cm $^{-1}$, an e^{-2} Diameter of 70 cm, an Absorption Coefficient of 0.07 km $^{-1}$, and No Scattering

Focal Length F (km)	Area A Containing 63% of the Focal-Plane Power (cm ^{2})	Relative Area A_{rel} Relative To No Thermal Blooming	Peak Intensity I_{peak} (kW/cm ^{2})	Relative Peak Intensity I_{rel} Relative To No Thermal Blooming
First Case: 10 MW Power, 250-m/s Wind, and No Slew				
2	57.6	20.3	147	0.0464
5	658	37.0	10.3	0.0251
10	3543	49.8	1.33	0.0184
Second Case: 2 MW Power, 2-m/s Wind, and 0.02-s $^{-1}$ Slew				
2	64.8	22.8	26.8	0.0422
5	474	26.6	2.96	0.0359
10	2018	28.4	0.495	0.0341

Another example of SSPARAMA output is illustrated in Fig. 3, namely, the final contour plot for the 5-km run from the first case with 10% contour levels. The complete printed output from SSPARAMA is included in Figs. 4a through 4c.

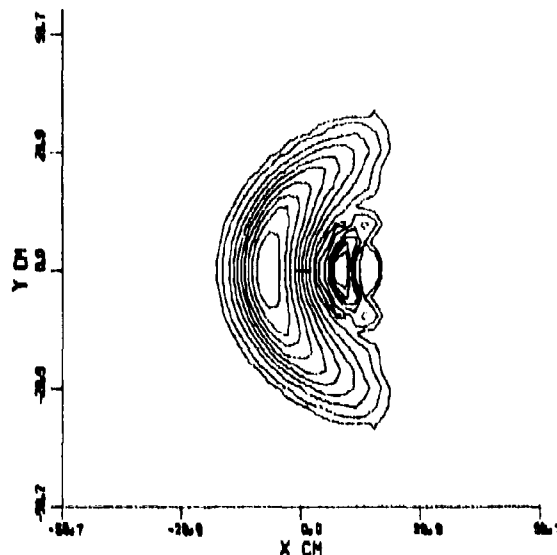


Fig. 3--Contour plot with 10% contour levels for the 5-km run from the first case in Table 4 (PNALF = 0.350, PNK = 10.400, PNO = 0.002, PNS = 0.000, PND = 80.000)

Figure 4a, the first page of printed output, is almost self-explanatory. Both dimensionless and physical parameters are listed; one is computed from the other, depending on which was entered. The program options indicate the mode, either CW or MP and the beamshape etc. The results summary in Fig. 4a includes the final value of the energy conservation integral, Eq. (2). This quantity, which is ideally equal to 1, gives a quick check on the validity of the numerical calculations. One factor that limits the accuracy is the use of a finite mesh size. As this mesh is made finer, the intensity distribution gets closer to the mesh boundaries, and numerical errors may enter through diffraction and the use of a discrete Fourier-transform routine as energy is reflected off the boundary. To avoid this reflection, the outermost boundary of the computational grid is set to zero and the next outermost boundary is set to one half its value at each z step. Thus the sum over normalized intensity gives an indication of how much energy was lost due to boundary-value problems.

The area that is given in Fig. 4a is the area containing exactly 0.63 of the total flux obtained by linear interpolation between adjacent flux fractional areas. This area will include contributions from several peaks as the intensity pattern breaks up under severe blooming conditions, so its meaning may also require a suitable interpretation of the intensity contour map. In addition the relative area and maximum intensity are calculated relative to the focal area and intensity of a vacuum-propagated infinite Gaussian whose e^{-1} diameter is equal to the value of WIDTH regardless of the beamshape being propagated.

WHITNEY, MADER, AND ULRICH

```

*** INPUT DATA ***

: INTERNAL PARAMETERS      PHYSICAL PARAMETERS      NUMERICAL PARAMETERS

AA = 0.000000             RADIUS(M) = 0.1240             HZ = 0.20
ALT = 0.350               ALPHA1(M) = 0.070000             HZ = 0.20
AM = 10.40               HZ(CM) = 6666.63             HZ = 0.20
AS = 0.00                V(HZ) = 644.98             HZ = 0.20
AT = 0.00                ENERGY(M) = 0.0000             PHIMAX = 1.000
AZ = 1.0000              ENERGY(M) = 1000.00
                                FWHM = 0.0000
                                HY(SEC) = 1.000000
                                ALPHA2(M) = 0.000000

PROGRAM OPTIONS

MODE                       INFINITE GAUSSIAN
BEAMSHAPE                  YES
ADAPTATION                 YES
HALF-STEP INTEGRATION      YES
PULSED CARD OUTPUT         NO
NUMBER OF PLOTS            5
LEG LEVEL CONTROLS         NO
RESCALE FINAL CONTOUR PLOT YES
  
```

*** RESULTS ***

```

THE CALCULATIONS REACHED z = 9.00000 (MM)
THE SUP OVER NORMALIZED INTENSITY = 1.00000
THE NUMBER OF Z-STEP = 20
AVERAGE POWER (KW) SHIPPED AT APERTURE = 10492.694
AVERAGE TRANSMITTED POWER (KW) = 7143.339
AREA (SQCM) CONTAINING 0.63 OF PDMH = 647.995
A REL RELATIVE TO INF. GAUSSIAN = 36.982
AVERAGE INTENSITY (KW/SQCM) IN THIS AREA = 7.031
PEAK INTENSITY (KW/SQCM) = 10.340
REL RELATIVE TO INF. GAUSSIAN PEAK = 0.02927
  
```

Fig. 4a—First page of the output by SSPARAMA, containing the input that resulted in Fig. 3 and a summary of the results

Figure 4b, the page containing numerical data, begins with a list of internally computed quantities that relate to the problems of air breakdown and t -cubed self-blooming. They are printed only for possible future data analysis. Assuming the breakdown intensity at $10.6 \mu\text{m}$ is $3 \times 10^6 \text{ W/cm}^2$ and that this is inversely proportional to wavelength squared, the following quantities are computed as a function of range: the minimum area required for breakdown, the ratio of this minimum area to the vacuum area, the maximum pulse length before breakdown occurs, the critical power, the saturation time, the intensity produced by the critical power propagating in a vacuum, and factors accounting for turbulence with values of C_n^2 of 10^{-15} and 10^{-14} . This is followed by an x and y slice through the aperture to check the initial beamshape.

The quantities, including the values of HZN in z/ka^2 units, relating to the coordinate system adaption are printed at each z step. The headings D, D1, D2, ALPHA1, ALPHA2, BETA1, DALPH1, DALPH2, DBET1, and XCEN correspond to D , D_1 , D_2 , α_1 , α_2 , β , $\Delta\alpha_1$, $\Delta\alpha_2$, $\Delta\beta$, and X used in the second section of this report. Also included is EPSMX, the maximum value of the summation given in Eq. (72); PHIMX, the maximum value of the positive phase change applied to ψ to obtain Φ ; and PARM, the number of pulses, for the MP mode, that occur in a computational cell.

Figure 4c, the output data, lists in the top portion the area, flux, the area fraction, and flux fraction contained within each contour level. From these data the 63% area is interpolated. This is followed in the middle portion by the z locations of the maximum of the average and peak intensities, the minimum 63% area, and the minimum z step that

*** NUMERICAL DATA ***

Fig. 4b—Second page of the output, containing numerical data

Summary of Program Structure

The structure of the code SSPARAMA is explained below and summarized in the flow chart in Fig. 5.

- 23

WHITNEY, MADER, AND ULRICH

*** OUTPUT DATA ***

LEVEL	AREA (SQ CM)	FLUX LMW	AREA FRACTION	FLUX FRACTION	RADIANCE (MW/SQ-CM)
0.0000	0.1781+001	0.019+002	0.0090	0.1173	0.011+000
0.0000	0.1784+002	1.030+003	0.1111	0.1219	0.000+000
0.0000	0.1784+002	3.354+003	0.1209	0.1219	0.000+000
0.0000	0.1784+002	3.374+003	0.1209	0.1219	0.000+000
0.0000	0.1784+002	4.261+003	0.1209	0.1219	0.000+000
0.0000	0.1784+002	4.915+003	0.1209	0.1219	0.000+000
0.0000	0.1784+002	5.784+003	0.1209	0.1219	0.000+000
0.0000	0.1784+002	6.383+003	0.1209	0.1219	0.000+000
0.0000	0.1784+002	6.920+003	1.0000	0.1219	0.000+000
0.0000	0.1784+002	1.144+003	1.1000	0.1219	0.000+000

MAXIMUM AVG I	= 1.873+001	AT Z	= 1.349+000
MAXIMUM PEAR I	= 1.841+001	AT Z	= 1.349+000
MINIMUM AREA	= 2.741+002	AT Z	= 1.349+000
MINIMUM RZ	= 2.176+003	AT Z	= 1.349+000

Z	IAVE	403	IMAX	MPMAX	VPMAX
0.000	3.481	1939.1802	3.308	0.000	0.000
0.000	3.488	1782.1802	3.310	0.000	0.000
0.000	3.473	1442.1809	4.139	0.000	0.000
0.000	3.477	1404.1790	6.108	0.000	0.000
0.000	4.074	1322.439	7.083	0.000	4.143
0.000	4.070	1225.445	7.870	0.000	3.972
0.000	5.201	1120.431	8.062	0.000	3.842
0.000	5.798	1022.437	8.833	0.000	3.684
0.000	6.274	900.996	9.217	0.000	3.588
0.000	6.791	868.196	9.329	0.000	3.500
0.000	7.430	784.193	10.772	0.000	3.497
0.000	8.200	677.193	12.064	0.000	3.499
0.000	9.120	591.193	13.519	0.000	3.133
0.000	10.786	503.194	16.884	0.000	3.086
0.000	12.000	414.197	20.707	0.000	3.140
0.000	13.787	329.210	25.811	0.000	3.108
0.000	15.787	244.196	31.820	0.000	3.137
0.000	18.001	161.191	38.413	0.000	3.196
0.000	20.509	91.191	45.413	0.000	3.203
0.000	23.270	37.194	51.340	0.000	3.203
0.000	26.239	4.6166	56.190	0.000	3.000
0.000	29.277	99.1900	10.499	0.000	0.000
0.000	32.031	607.199	10.349	0.000	0.000

Fig. 4c—Third page of the output, containing the remaining numerical data

- The initialization procedure continues with the call to INTENS, where the aperture intensity is computed at each mesh point.
- The call to DENS computes the quantity $g(x, y, z)$ given in Eq. (63) and then applies the phase change given by Eq. (62) which converts ψ to Φ . The first z increment is also computed.
- The main program loop begins here with a call to OUTPUT to store various values until the calculations are completed.
- The call to ADVANCE applies the Fourier transform of Eq. (67) and then the phase change of Eq. (68). The array is Fourier-transformed back to yield $\Phi(z + \Delta z)$.
- The intensity is computed with the call to INTENS, and the boundary values of the array are tapered to zero.
- The call to DENS now includes a call to VTRANS, by which the phase change of Eq. (62) is reversed, converting Φ back to ψ . The quantities $\{\alpha_1, \alpha_2, \beta\}$ and $\{\Delta\alpha_1, \Delta\alpha_2, \Delta\beta\}$ are found in VTRANS, and the values of D_1 and D_2 are updated. After the return to DENS, Eq. (63) is solved and the phase change of Eq. (62) is reapplied, converting ψ back to Φ in preparation for the next call to ADVANCE.

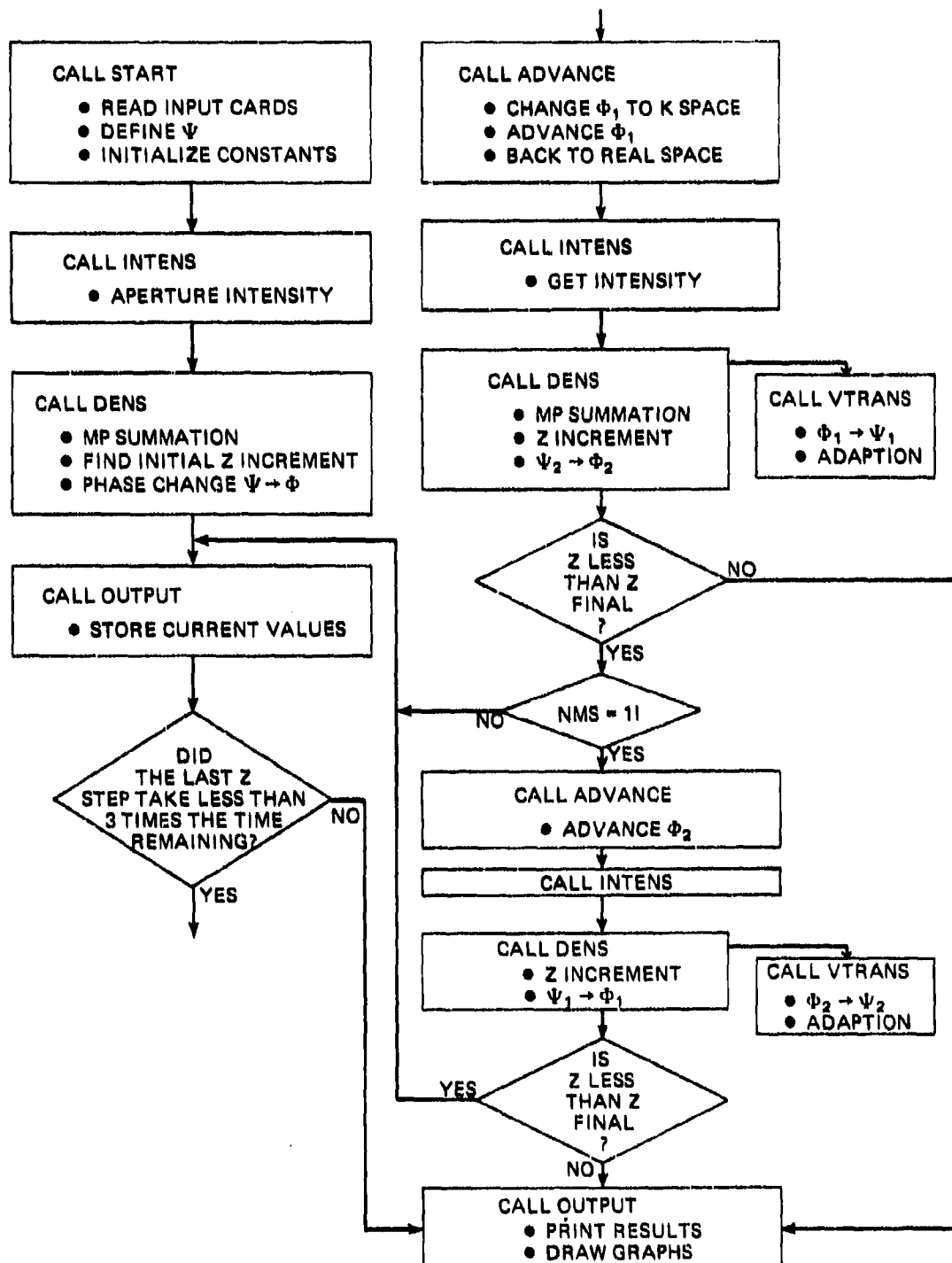


Fig. 5—Summary of the code SSPARAMA

WHITNEY, MADER, AND ULRICH

- Now that one cycle of propagating the solution is completed, the code checks if z final has been reached and if the half-step integrations are to be performed as outlined in the section titled Numerical Procedures.

- When z final has been reached or the time limit of execution is near, the last call to OUTPUT prints the results and ends this run.

The Appendix contains a complete listing of the code with copious comments included.

REFERENCES

1. P.B. Ulrich, J.N. Hayes, J.H. Hancock, and J.T. Ulrich, "Documentation of PROP E, a Computer Program for the Propagation of High Power Laser Beams Through Absorbing Media," NRL Report 7681, May 29, 1974.
2. P.B. Ulrich, "PROP-I: An Efficient Implicit Algorithm for Calculating Nonlinear Scalar Wave Propagation in the Fresnel Approximation," NRL Report 7706, May 29, 1974.
3. 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.
4. J. Wallace and J.Q. Lilly, "Thermal blooming of repetitively pulsed laser beams," *J. Opt. Soc. Am.* 64, 1651-1655 (Dec. 1974).
5. J. Herrmann and L.C. Bradley, "Numerical Calculation of Light Propagation," Massachusetts Institute of Technology, Lincoln Laboratories, Lexington, Mass. Report LTP-10, July 12, 1971.
6. H.J. Breaux, "An Analysis of Mathematical Transformations and a Comparison of Numerical Techniques for Computation of High Energy CW Laser Propagation in an Inhomogeneous Medium," BRL Report 1723, June 1974.
7. K.G. Whitney and P.B. Ulrich, "Scaling Laws for Multipulse Steady-State Thermal Blooming," NRL Memorandum Report 3229, Mar. 1976, Appendix B3.

BEST AVAILABLE COPY

APPENDIX A

Listing of Code and Comments

```

PROGRAM SSPARMA
C * * * * *
COMMON /999/ A1(64,64), A2(64,64), ITENS(64,64)
COMMON /AAA/ EPS(64,64), EPS0(64,64), ACUT(11,99), BOUT(9,10),
  • AIN(2,64), DALPH1(2), DALPH2(2), DELT1(2), ALPH10(2), ALPH20(2),
  • RET10(2), D10(2), D20(2), RD10(2), RD20(2), SRTD10(2), XCEN0(2)
COMMON /BBB/ TENS(64,64), G1(64), G2(64), PHAS1(64), PHASE2(64),
  • CONMIN(10), M2(3), SV1(64), SV2(64), PARM(80)
COMMON /SINGLS/ F, PNA, PNALE, PNAK, PNO, PNS, PND, PNZ, HX, HY,
  • HZ, Z, ZZ, ZZH, ZNM, ZFINAL, XZERO, YZERO, XDT0, ALPHA, KN,
  • VDT, OMDT, HT, ENERGY, ALPHAC, CS, REFRC, GAMMA, FTO, CTK,
  • LTKJ, RHT, POUT, DARFA, T2, TS, TPULSE, AS2, PCR, SI, TCON1, TCON2,
  • Z1, R163MX, Z2, R163MX, Z3, APMN, Z4, HZM, DKARL, TENS MX,
  • EX, PHIMX, EPSMX, ERPMX, DGMX, R1, RDI MAX, VTERM, PHIFXX, HZNN5,
  • P1, IMAX, JMAX, NX, NY, NAD, NX2, NY2, NXY, NXDIN, NYDIN, NPT,
  • IPLOT, NITER, NBUF, NX1, NY1, NPS, FLAG, D, L1, D2, P1, P2, SRTD1, SRTD2,
  • RSRD12, XCEN, TLAST, SORTB, PND, CCOS0, CCOS1, D1, HCZ10, HCZ20, HCZ1X,
  • HCZ2X, HCZ12, ALPH1, ALPH2, RET1, CON1, CON2, HZ0, HZ1, LXC, EXN, T1, T2
COMMON /OUTS/ NNA, SCL, FAC, NKS, NPP, NCA, NEXIT, NPLOT, NPUNCH
C * * * * *
COMPLEX A1, A2
LOGICAL L5
DATA (CF=34, CH, ) , (REFRAC=0.154), (GAMMA=1.4), (FTJ=1.0E-7),
  • (CTK=1.0E-5), (PI=3.14159265), (ND1=1, CF6), (DTEJ=1.0E-3)
DATA (NXDIR=64), (NYDIN=64), (HZ=20)
DATA (I=1)/999/

C
C
C INPUT AND INITIALIZATION
C
ISTART=TIMELEFT(JOBBY)
L5=.FALSE.
CONTINUE
CALL START(L5)
NXTI=1
NITER=0
IPLOT=0
ZZ1=0.0
ZZ2=0.0
I2=2
IF (MPS .EQ. 0) I2=1

CALL INIT5(A1, A2, L5)
CALL TENS(A1, A1, ZZ1, 1, 1, .FALSE.)
HZ0=0.0

C
C *****
C
C MAIN PROGRAM LOOP
C
14 CONTINUE
NITER=NITER+1

C
C STORE VALUES FOR LATER PRINTOUT
C
CALL OUTPUT(.FALSE.)

C
C IF TIME REMAINING IS LESS THAN 3 TIMES THAT FOR THE LAST
C 2 STEP - EXIT
C

```

BEST AVAILABLE COPY

WHITNEY, MADER, AND ULRICH

```

3  TNOW=TIME(LEFT)
   DT=TLAST-TNOW
   TLAST=TNOW
   IF (3*DT.LE.TNOW) GO TO 8
   PRINT 27,Z
22  FORMAT(/25X25H*** TIME APORT AT Z(K) =11.5.1X3H***//)
   GO TO 13
8  CONTINUE
C
C  ADVANCE FROM ZZ TO ZZ+DZ, CALCULATING NEW ALTITUDES IN A
C
40  CALL ADVNCE(A1,1)
   CALL INTENS(A1,.FALSE.)
   CALL DENS(A1,A2,ZZ1,1,12,.TRUE.)
   IF (Z.GE.ZFINAL) GO TO 15
C
C  REPEAT IF HALF-STEP INTEGRATION IS INCLUDED
C
   IF (NMS.EQ.1) GO TO 45
   CALL ADVNCE(A2,2)
   CALL INTENS(A2,.FALSE.)
   CALL DENS(A2,A1,ZZ2,2,1,.TRUE.)
   IF (Z.GE.ZFINAL) GO TO 15
45  CONTINUE
   GO TO 14
C
C *****
C
C  SET NEXT EQUAL 1 FOR PRELIMINARY EXITS
C
13  NEXT=1
15  CONTINUE
C
C  EXECUTE ALL OUTPUT
C
   CALL OUTPUT(.TRUE.)
   PRINT 16
16  FORMAT(1H1)
17  CALL STOPPLOT
C
   PRINT RUN TIME (CP TIME).
   TRUN=(TSTART-TIME(LEFT))/.001/60.
   PRINT 18,TRUN
18  FORMAT(/2.16(10H)* RUN,11.1=*,0.6,2.5* PRINT,5H)

   STOP
   END

```

SUBROUTINE START(LS)

[illegible]

100-443886-100

```
26 IF (NCW.EQ.1) HT=1.
```

31

EST AVAILABLE COPY

WHITNEY, MADER, AND ULRICH

```

IF (NAD.EQ.0) PRINT 152
IF (NAD.EQ.1) PRINT 153
IF (NMS.EQ.0) PRINT 154
IF (NMS.EQ.1) PRINT 155
IF (NPUNCH.EQ.0) PRINT 156
IF (NPUNCH.EQ.1) PRINT 157
PRINT 158, NPLOT
IF (NCT.EQ.0) PRINT 162
IF (NCT.NE.0) PRINT 163
IF (NRS.EQ.0) PRINT 164
IF (NRS.NE.0) PRINT 165
148 FORMAT(20X9HBEAMSHAPE5X*INFINITE GAUSSIAN*)
149 FORMAT(20X9HBEAMSHAPE5X*TRUNCATED GAUSSIAN*)
150 FORMAT(20X9HBEAMSHAPE5X*UNIFORM CIRCLE*)
151 FORMAT(20X9HBEAMSHAPE5X*UNIFORM SQUARE*)
152 FORMAT(21X8HADAPTION5X2HNO)
153 FORMAT(21X8HADAPTION5X3HYF5)
154 FORMAT(1X21HHALF-STEP INTEGRATION5X2HNO)
155 FORMAT(1X21HHALF-STEP INTEGRATION5X3HYF5)
156 FORMAT(10X19HPUNCHED CARD OUTPUT5X2HNO)
157 FORMAT(10X19HPUNCHED CARD OUTPUT5X3HYF5)
158 FORMAT(14X19HNUMBER OF PLOTS5X13)
162 FORMAT(11X18HLOW LEVEL CONTOURS5X2HNO)
163 FORMAT(11X18HLOW LEVEL CONTOURS5X3HYF5)
164 FORMAT(13X26HRESALE FINAL CONTOUR PLOTS5X2HNO)
165 FORMAT(13X26HRESALE FINAL CONTOUR PLOTS5X3HYF5)
168 FORMAT(20X9HBEAMSHAPE5X*UNIFORM CIRCLE - OCCULTED RADIUS **F5.2)
C
C*****
C
C DEFINE AND STORE COMPARATIVE PHYSICAL DATA
C
C13=1.0/3.0
W2=WIDTH*WIDTH
PL=(2.0*PI/ZN)*1.0 F4
SD1=SD1*(1.0/PL)**2
RHT=1.0/HT
POUT=ENERGY*RHT*EJTCJ
DARE=HXX*HY
CASE=SQRT(ALPHA*ENERGY*EJTCJ*4/19.0)*6*PI*W2*1.0*1.0*1.0**2.1
CTP=3.0 F=4*SQRT(222*ALPHA*ENERGY*EJTCJ)
CPCR=(1.0/CTP-0.0001)*0.01*(1.0/CTP-1.0)*ALPHA*W2*1.0*1.0*1.0**2.1
ZF1NAL=ZF1*CTK*0.000000
ZF1=PNZ/PNK
TPULSE=PT
NP=10
C
DO 56 I=1,NP
ZN=FLOAT(I)-1)/FLOAT(I)-1)+1.0**3
ZETA=ZN/PNK
Z=ZETA*F*CTK
U=ZETA*ZETA*(1.0-ZN)**2
AV=U/2.0
EX=(XP-PAALF*ZN)
SREX=SQRT(EX)
AS4=CASE*Z*F*SREX**3
AS2=SQRT(AS4)
AR=AS2/AV
PCR=CPCR*AS4**C13/ZX
TP=CTP*AS2/(SREX*ZETA*(1.0-6.0))
TPULSE=AMIN1(TPULSE,TP)
TS=PCR*TP/POUT
SI=0.75*PCR*EX/(PI*AV)
IF (Z.NE.0.00) GO TO 10
TCOR1=1.0
TCOR2=1.0

```

NRL REPORT 8074

BEST AVAILABLE COPY

```

10      GO TO 12
      DTURB1=DTURBF(1.0E-15)
      TCOR1=D/DTURB1
      DTURB2=DTURBF(1.0E-14)
      TCOR2=D/DTURB2
12      CONTINUE
      ROUT(1,1)=Z
      ROUT(2,1)=AS2
      ROUT(3,1)=AR
      ROUT(4,1)=TP
      ROUT(5,1)=PCR
      ROUT(6,1)=TS
      ROUT(7,1)=S1
      ROUT(8,1)=TCOR1
      ROUT(9,1)=TCOR2
94      CONTINUE
C *****
C
C      DEFINE INITIAL AMPLITUDES AT APERTURE
C
C      TRUNCATED OR INFINITE GAUSSIAN
C      UNIFORM CIRCLE OR SQUARE
C
C      TRUNCATED GAUSSIAN IS TRUNCATED AT 1/E INTENSITY RADIUS
C      OR R(TRUN.)=1.414**A
C
      XZERO=(NX-1)*HX/2.
      YZERO=(NY-1)*HY/2.
      NXM=NX-1
      NYM=NY-1
      DKAREA=NX*NY*DKARF
      DO 64 J=1,NY
      Y=(J-1)*HY+YZERO
      G2(J)=1.0-Y*Y
      DO 64 I=1,NX
      X=(I-1)*HX+XZERO
      IF (J.EQ.1) G(1)=1.0-X*Y
      SSG=XX*Y+Y*Y
C
C      DEFINE GAUSSIAN AMPLITUDE
C
      IF (NM.GT.2) GO TO 310
      REAL=XP(-.5*SSG)
      IF (NM.EQ.1.AND.SSG.GT.2.0) REAL=.0
      GO TO 350
C
C      DEFINE UNIFORM CIRCLE AMPLITUDE
C
      IF (NM.GT.2) GO TO 310
      REAL=1.
      IF (SSG.GT.1.0) REAL=.0
      GO TO 350
C
C      DEFINE SQUARE APERTURE
C
      IF (NM.GT.3) GO TO 320
      IF (ABS(X).LE.1. .AND. ABS(Y).LE.1.0 .AND. NM.EQ.3) REAL=1.
      IF (ABS(X).GT.1.0 .OR. ABS(Y).GT.1.0 .AND. NM.EQ.3) REAL=.0
      GO TO 350
C
C      DEFINE OCCULTED UNIFORM CIRCLE
C
      IF (NM.GT.1) GO TO 320
      IF (SSG.GT.1.0) REAL=.0
      IF (SQRT(SSG).LE.ROCU) REAL=.0

```

WHITNEY, MADER, AND ULRICH

BEST AVAILABLE COPY

```

C
C   LOAD INITIAL ARRAY
C
350  CONTINUE
    A1(I,J)=CMPLX(REAL,0.0)
64   CONTINUE
C
C   FIND NORMALIZATION FACTOR
C
    CALL INTENS(A1, .FALSE.)
    RNORM=0.0
    DO 66 J=1,NY
    DO 66 I=1,NX
        RNORM=RNORM+TENS(I,J)
66   CONTINUE
    RRNORM=1./ZSQRT(RNORM*DAREA)
C
C *****
C
C   INITIALIZE CONSTANTS AND PARAMETERS USED LATER IN PROGRAM
C
    NX2=2*NXDIM*NX
    NY2=2*NYDIM*NYDIM
    NY2=NY2/2
    PV(1)=2.0*PI*(1-1)/(NX*NX)
    IL=NX2+1
    DO 70 I=1,IL,NX
        PV(1)=2.0*PI*(1-1-IX)/(NX*NX)
        PV(1)=PV(1)*PV(1)
70   DO 72 J=1,NY
        PHASE2(J)=.5*PV(1)
    DO 72 I=1,NX
C
C   NORMALIZE APPLICABLE TO THIS LOOP
C
        A1(I,J)=RRNORM*A1(I,J)
        A2(I,J)=A1(I,J)
        EPS1(I,J)=.
        IF (J-1-NY-1) PHASE1(I,J)=.5*PV(1)
72   CONTINUE
    DO 74 J=1,NY
        EPS1(I,J)=.
74   IF (I-1-NX-1) PHASE1(I,J)=.
C
C   STORE X AND Y COORDINATES OF APERTURE
C
    IPX=NX/2
    IPY=NY/2
    DO 62 I=1,66
        A1(I,1)=A1(IPX,1)
62   A1(I,2)=A1(1,IPY)
C
        B1=APX+.5
        B2=APX+.5
        APX=1.0/1.
        BZ=1.0/1.
        BZ=0.0
        COM1=.5*(1.-1.)/ZSQRT(1.-1)
        COM2=.5*(1.+1.)/ZSQRT(1.-1)
        P2(1)=LOGE(1.-BX)/ZLOGE(2.)+.5
        P2(2)=LOGE(1.-BY)/ZLOGE(2.)+.5
        P2(1)=
        CALL SETUP (1, P2, PV, NY2, ., 1, 1, 1)

```

NRL REPORT 8074

[illegible]

WHITNEY, MADER, AND ULRICH

```

SUBROUTINE ADVNCE(A,NS)
C
C THIS SUBROUTINE FOURIER TRANSFORMS PHI(X,Y,Z) TO PHI(K1,K2,Z),
C AND ADVANCES THE SOLUTION BY APPLYING THE PHASE CHANGE
C
C PHI(K1,K2,Z+HZ)=PHI(K1,K2,Z)*EXP(0.5*I*(K1**2*HZ/D1+K2**2*HZ/D2))
C
C AND THEN TRANSFORMS BACK TO REAL SPACE
C
C *****
COMMON /AAA/ EPS(64,64),EPSO(64,64),AOUT(11,99),BOUT(9,10),
  AIN(2,64),DALPH1(2),DALPH2(2),DBET1(2),ALPH10(2),ALPH20(2),
  BET10(2),D10(2),D20(2),RD10(2),RD20(2),SRD10(2),XCENO(2),
  COMMON /BBB/ TENS(64,64),G1(64),G2(64),PHASE1(64),PHASE2(64),
  CONMIN(10),M2(3),SV1(64),SV2(64),PARM(80)
COMMON /SINGLS/ F, PNA, PNALF, PNK, PNO, PNS, PND, PNZ, HX, HY,
  HZ, Z, ZZ,ZZF, ZNM, ZFINAL, XZERO, YZERO, WIDTH, ALPHA, WN,
  VODT, OMDT, HT, ENERGY, ALPHAC, CS, REFRAC, GAMMA, ETJ, CTJ,
  EUTKJ, RHT, POUT, DAREA, X2, TS, TPULSE, AS2,PCR,S1,TCOR1,TCOR2,
  Z1, R163MX, Z2, R163MX, Z3, APMN, Z4, HZMN, DKAREA, TENSIX,
  EX, PHIMX, EP5MX, FRRMX, DGMX, R1, BDIMAX, VTERV, PHIMXX,HZNS,
  P1, IMAX, JMAX, NX, NY, PAD, NX2, NY2, NXY, NXDIM, NYDIM, NPT,
  IPLOT, NITER, NBUF, NXX, NYN, NMS, NFLAG, D, D1, D2, P1, P2, SRD1, SRD2,
  RSRD1, XCEN, TLAST, SURT8, PNDF, GCON0, GCON, GDI, HCZ10, HCZ20, HCZ1N,
  HCZ2N, HCZ12, ALPH1, ALPH2, BET1, CON1, CON2, HZ0, HZN, EXO, EXN, T1, T2
C *****
COMPLEX A(64,64)
C
C DEFINE PARAMETERS FOR PHASE TRANSFORMATION
C
  HZ1=CON1*HZ0
  HZ2=CON2*HZN
  ZD11=(HZ1+HZ0)*RD1/(NS)
  ZD12=(HZ2+HZ0)*RD1/(NS)
  ZD21=(HZ1+HZ0)*RD2/(NS)
  ZD22=(HZ2+HZ0)*RD2/(NS)
  D11=D1/(NS)* ((1.+2.*ALPH1/(NS))* ZD11)**2+ZD11*ZD11
  D12=D1/(NS)* ((1.+2.*ALPH1/(NS))* ZD12)**2+ZD12*ZD12
  D21=D2/(NS)* ((1.+2.*ALPH2/(NS))* ZD21)**2+ZD21*ZD21
  D22=D2/(NS)* ((1.+2.*ALPH2/(NS))* ZD22)**2+ZD22*ZD22
  RD11=1./D11
  RD12=1./D12
  RD21=1./D21
  RD22=1./D22
  HCZ1N=0.5*HZN*(RD11+RD12)
  HCZ2N=0.5*HZN*(RD21+RD22)
  RSRD51=SQRT(RD11*RD21)
  RSRD52=SQRT(RD12*RD22)
  HCZ12=0.5*HZN*(RSRD51+RSRD52)
C
C WHEN NMS=0, P2=0
C
  W11=P2*(HCZ12-HZ1*RSRD51-HZ2*RSRD52)
  W12=HCZ12-W11
C
C RESORT ARRAY IF NX LT 64
C
  IF (NX.EQ. NXDIM) GO TO 1155
  DO 115 J=1,NY
  DO 115 I=1,NX
115 A((J-1)*NX+I)=A(I,J)
1155 CONTINUE

```

14

NRL REPORT 8074

```

C
C PERFORM FOURIER TRANSFORM - TO K-SPACE
C
C   CALL FASTFOUR(A(1,1), M2, SV1, SV2, -1, IFERR)
C   IF (NX .LT. NXDIM) GO TO 10
C
C PLOT FOURIER TRANSFORM OF INTENSITY DISTRIBUTIONS AT
C APERTURE AND Z=FINAL
C
C   IF (NITER .EQ. 1 .AND. NS .EQ. 1) CALL INTENS(A, .TRUE.)
C   IF (ZZ+MZN+HZNMS.EQ.ZZF) CALL INTENS(A, .TRUE.)
10  CONTINUE
C
C APPLY PHASE CHANGE TO ADVANCE THE CALCULATIONS
C
C   DO 12 J=1,NY
C   DO 12 I=1,NX
C     JT=(J-1)*NX+I
C     PHI=PI*((HCZ10+HCZ1N)*PHASE1(I)+(HCZ20+HCZ2N)*PHASE2(J))
12  A(JT)=A(JT)*CMPLX(COS(PHI),-SIN(PHI))
C
C PERFORM FOURIER TRANSFORM - TO REAL-SPACE
C
C   CALL FASTFOUR(A(1,1), M2, SV1, SV2, 1, IFERR)
C
C RESORT ARRAY IF NX LT 64
C
C   IF (NX .EQ. NXDIM) GO TO 1165
C   DO 116 J=1,NY
C   DO 116 I=1,NX
116  A(I,J)=A(I+(J-1)*NX)
1165 RETURN
END

```

BEST AVAILABLE COPY

WHITNEY, MADER, AND ULRICH

```

SUBROUTINE DENS(A, B, ZCOORD, ND1, ND2, LD)
C
C THIS SUBROUTINE APPLYS THE PHASE CHANGE
C
C (1-X**2)/D1+(1-Y**2)/D2-3NIGAMMA-1)*K**2*E/C5**2/SQRT(D1*D2)*
C SUM(PHI(X-XP,Y,Z))**2
C
C THIS CONVERTS PSI (OR A) TO PHI
C
C * * * * *
COMMON /AAA/ EPS(64,64),EPS0(64,64),AOUT(11,99),BOUT(9,10),
  AIN(2,64),DALPH1(2),DALPH2(2),DBET1(2),ALPH10(2),ALPH20(2),
  BET1(2), D10(2), D20(2), RD10(2), RD20(2), SRTD10(2), XCEN0(2)
COMMON /HHH/ TEN5(64,64), G1(64), G2(64), PHASE1(64), PHASE2(64),
  CONIN(10),X2(3),SV1(64),SV2(64),PARM(80)
COMMON /SINGLS/ F, PNA, PNALE, PNK, PRO, PNS, PND, PNZ, PX, PY,
  HZ, Z, ZZ,ZZF, ZNM, ZFINAL, XZERO, YZERO, WIDTH, ALPHA, WN,
  VDET, OMDT, HT, ENERGY, ALPHAC, CS, RIFRAC, GAMMA, ETJ, CTK,
  ETKJ, RHT, POUT, DAREA, H2, TS, TPULSE, AS2,PCR,SI,TCOR1,TCOR2,
  Z1, H163MX, Z2, R1MXIX, Z3, APIN, Z4, HZMN, DKAREA, TEN5IX,
  EX, PHIMX, EPSMX, ERMX, DOMX, R1, BDIMAX, VTERM, PHIMX, HZMS,
  P1, IMAX, JMAX, NX, NY, NAD, NX2, NY2, NXY, NXDIM, NYDIM, NPT,
  IPLOT, NITER, NBUF, NXM, NYM, NKS, NFLAG, D1, D2, P1, P2, SRTD1, SRTD2,
  RSKD12, XCEN, TLAST, SQRTB, PND, GCON0, GCON, H01, HC210, HC220, HC21N,
  HC22N, HC212, ALPH1, ALPH2, BET1, CON1, CON2, HZ0, HZK, X0, EXN, W11, W12
COMMON /OUT5/ HNM, SCLFAC, NKS, NPP, HCE, HEXIT, NPLOT, NPUNCH
C * * * * *
COMPLEX A(64,64), B(64,64)
LOGICAL LD
DIMENSION TEN (10), EPS0(10), HZSAVE(2)
C
C INITIALIZE 7-STEP ON FIRST CALL
C
C IF (LD) GO TO 41
HZNM=0.
HZNM5=0.0
HZSAVE(1)=0.
HZSAVE(2)=0.
P1=0.5
P2=0.0
P3=1.
EXN=1.
C
C ZZF = ZFINAL / K * WIDTH**2
C
C HZMX=0.1**ZZF
C IF (NMS,FO) HZMX=0.5**HMX
C HZMINI=1.0**4*ZZF
C IF (NMS,FO) HZMINI=0.5**2*HMI
41 ZCOORD=ZCOORD+HZSAVE(1)
C
C ZZ = ZICM1 / X * WIDTH**2
C
C ZZ=ZCOORD
C
C ZNM = Z / F
C
C ZNM=ZZ*PNK
C
C Z = Z(KM)
C
C Z=ZNM*F*(TK+1.0)-b
C
C D=ZZ*ZZ+(1.0-ZNM)**2
ZD1=HZ*RD1. (ND1)

```

```

D1=D10(ND1)*((1.0+2.0*ALPH10(ND1)*ZDD1)**2+ZDD1*ZDD1)
SRTD1=SQRT(D1)
C
C VTERM = DISTANCE BETWEEN PULSES / DISTANCE BETWEEN GRID POINTS
C
VTERM=2.0*(1.0+PNS*ZNM)/(HX*SRTD1*PNO)
IF(VTERM.LT.0) GO TO 45
EPSMX=0.0
JC=FIX(1.0/VTERM)+1
DC=FLOAT(JC-2)
IF (NCW.EQ.1) GO TO 200
IF (DC) 48,49,50
C
C EXIT OPTIONS
C
45 NPUNCH=0
NEXIT=1
PRINT 100, Z
FORMAT(//2X, *AT Z=, F6.4, * KM, A DEAD ZONE IS PRESENT IN THE C
ALCULATION*)
CALL OUTPUT(.TRUE.)
STOP
C
46 PRINT 101, Z
FORMAT ( // 2X *AT Z= * F6.4 * KM, THERE ARE MORE THAN 10 PULSES
PER CELL PRESENT IN THE CALCULATION*)
STOP
C
47 NPUNCH=0
NEXIT=1
PRINT 103, Z
FORMAT( // 2X *AT Z= * F7.4 * KM, THE CALCULATED IZ IS SMALLER TH
AN THE MINIMUM ALLOWED VALUE*)
CALL OUTPUT(.TRUE.)
STOP
C
C *****
C
C COM THE INTENSITY ACROSS THE GRID FOR MULTI-PULSE
C INTEGRATE THE INTENSITY ACROSS THE GRID FOR CW
C
C LESS THAN ONE PULSE PER CELL
C
48 I1=VTERM
I11=I1+1
F1=I11-VTERM
F2=1.0-F1
DO 4 J=1,NY
DO 4 I=2,NX
IF(I-I11.GE.1) GO TO 44
EPSO(I,J)=EPS(I,J)
EPS(I,J)=...
GO TO 4
44 EPSO(I,J)=EPS(I,J)
EPS(I,J)=F1*(TENS(I-I1,J)+EPS(I-I1,J))
F2*(TENS(I-I11,J)+EPS(I-I11,J))
IF (TENS(I,J) .GT. .05*TENSMX) EPSMX=AMAX1(PNS, EPS(I,J))
4 CONTINUE
GO TO 51
C
C ONE TO TWO PULSES PER CELL
C
49 UTERM=2.0*VTERM
F1=2.0-UTERM
F2=1.0-F1

```

WHITNEY, MADER, AND ULRICH

```

DO 5 J=1,NY
  EPS1=0.0
  TEN1=0.5*(TENS(1,J)+TENS(2,J))
  EPSO(2,J)=EPS(2,J)
  EPS(2,J)=F1*TEN1+F2*TENS(1,J)
  EPSMX=AMAX1(EPS(2,J), EPSMX)
DO 5 I=3,NX
  EPS1=F1*(TENS(I-1,J)+EPS(I-1,J))+F2*(TEN1+EPS1)
  TEN1=0.5*(TENS(I-1,J)+TENS(I,J))
  EPSO(I,J)=EPS(I,J)
  EPS(I,J)=F1*(TEN1+EPS1)+F2*(TENS(I-1,J)+EPS(I-1,J))
  IF (TENS(I,J) .GT. 0.05*EPSMX) EPSMX=AMAX1(EPSMX, EPS(I,J))
5 CONTINUE
  GO TO 51
C
C MORE THAN TWO PULSES PER CELL
C
50 IF (JC .GT. 10) GO TO 46
  UTERM=FLOAT(JC)*VTERM
  F1=2.0-UTERM
  F2=1.0-F1
DO 6 J=1,NY

```

BEST AVAILABLE COPY

NRL REPORT 8074

```

I=2
EPSO(1)=0.0
EPSO(2)=0.0
TENO(1)=TENS(I-1,J)
TENO(2)=(TENS(I,J)+(FLOAT(JC)-1.0)*TENS(I-1,J))/FLOAT(JC)
DO 70 JJ=3,JC
  FJ=FLOAT(JJ-1)/FLOAT(JC)
  TENO(JJ)=FJ*TENS(I,J)+(1.0-FJ)*TENS(I-1,J)
  EPSO(JJ)=F1*(TENO(JJ-1)+EPSO(JJ-1))+F2*(TENO(JJ-2)+EPSO(JJ-2))
70 CONTINUE
  EPSO(I,J)=EPS(I,J)
  EPS(I,J)=F1*(TENO(JC)+EPSO(JC))+F2*(TENO(JC-1)+EPSO(JC-1))
  EPSMX=AMAX1(EPS(I,J), EPSMX)
DO 6 I=3,NX
  EPSO(1)=EPS(I-1,J)
  TENO(1)=TENS(I-1,J)
  EPSO(2)=F1*(TENO(1)+EPSO(1))+F2*(TENO(JC)+EPSO(JC))
  TENO(2)=(TENS(I,J)+(FLOAT(JC)-1.0)*TENS(I-1,J))/FLOAT(JC)
DO 71 JJ=3,JC
  FJ=FLOAT(JJ-1)/FLOAT(JC)
  TENO(JJ)=FJ*TENS(I,J)+(1.0-FJ)*TENS(I-1,J)
  EPSO(JJ)=F1*(TENO(JJ-1)+EPSO(JJ-1))+F2*(TENO(JJ-2)+EPSO(JJ-2))
71 CONTINUE
  EPSO(I,J)=EPS(I,J)
  EPS(I,J)=F1*(TENO(JC)+EPSO(JC))+F2*(TENO(JC-1)+EPSO(JC-1))
  IF (TENS(I,J).GT.0.05*EPSMX) EPSMA=AMAX1(EPSMA,EPS(I,J))
6 CONTINUE
GO TO 51
C
C COMPUTE CW INTEGRAL
C
200 DO 110 J=1,NY
  EPSO(1,J)=EPS(1,J)
  EPS(1,J)=0.5*HX*TENS(1,J)*WIDTH/(VDT*(1.+PNS*ZNM))
  1 *SQRT(D1)
  DO 110 I=2,NX
    EPSO(I,J)=EPS(I,J)
    EPS(I,J)=EPS(I-1,J)+.5*HX*(TENS(I,J)+TENS(I-1,J))*WIDTH/
    1 (VDT*(1.+PNS*ZNM))
    1 *SQRT(D1)
    IF (TENS(I,J).GT.0.05*EPSMX) EPSX=AMAX1(EPSX,EPS(I,J))
110 CONTINUE
C
C *****

```

BEST AVAILABLE COPY

WHITNEY, MADER, AND ULRICH

```

C 51 EXO=EXN
    EX=EXP(-PNALF*ZNM)
    EXN=EX
    GCON=GCONO*EX
    IF (LD) CALL VTRANS(A,ND1)
    HZO=HZN
    HCZ1O=HCZ1N
    HCZ2O=HCZ2N

C
C
C CALCULATE THE Z-INTEGRALS WHEN NMS = 0

    IF (NMS.NE.0) GO TO 52
    HZNMS=HZO
    HZ1=CON1*HZO
    HZ2=CON2*HZO
    ZD11=HZ1*RD1O(ND1)
    ZD12=HZ2*RD1O(ND1)
    ZD21=HZ1*RD2O(ND1)
    ZD22=HZ2*RD2O(ND1)
    D11=D1O(ND1)*(11.0+2.*ALPH1O(ND1)*ZD11)**2+ZD11*ZD11
    D12=D1O(ND1)*(11.0+2.*ALPH1O(ND1)*ZD12)**2+ZD12*ZD12
    D21=D2O(ND1)*(11.0+2.*ALPH2O(ND1)*ZD21)**2+ZD21*ZD21
    D22=D2O(ND1)*(11.0+2.*ALPH2O(ND1)*ZD22)**2+ZD22*ZD22
    RD11=1./D11
    RD12=1./D12
    RD21=1./D21
    RD22=1./D22
    RSRDS1=SQRT(RD11*RD21)
    RSRDS2=SQRT(RD12*RD22)

C
C
C THE 3 Z-INTEGRALS

    HCZ1O=.5*HZO*(RD11+RD12)
    HCZ2O=.5*HZO*(RD21+RD22)
    HCZ12=.5*HZO*(RSRDS1+RSRDS2)
    RT2=HCZ12

C
C
C COMPUTE NEW Z INCREMENT IN ZCON*Z OF L15

52 HZN=H3*AM1(L15,4*H1+1,4*H2+1,PHIPXX)
    ZCON12=CONO*(PHIPX+1,PHIPX+1)
    IF (HZN.GT.HZPX) HZN=HZPX
    IF (HZN.LT.HZMIN1) GO TO 47
    IF (HZN.GT.Z1-Z2-HZNS) HZNS=Z1-HZNS*Z1-Z2+HZN
    HZ=HZO+HZN
    HZSAVE(ND1)=HZ

C
C
C COMPUTE THE THREE Z-INTEGRALS

C
C
C NO ONLY ON FIRST CALL

    IF (LD) GO TO 54
    IF (NMS.EQ.0) P3=.5
    HZ1=CON1*HZ
    HZ2=CON2*HZ
    ZD11=HZ1*RD1O(ND2)
    ZD12=HZ2*RD1O(ND2)
    ZD21=HZ1*RD2O(ND2)
    ZD22=HZ2*RD2O(ND2)
    D11=D1O(ND2)*(11.0+2.*ALPH1O(ND2)*ZD11)**2+ZD11*ZD11
    D12=D1O(ND2)*(11.0+2.*ALPH1O(ND2)*ZD12)**2+ZD12*ZD12
    D21=D2O(ND2)*(11.0+2.*ALPH2O(ND2)*ZD21)**2+ZD21*ZD21
    D22=D2O(ND2)*(11.0+2.*ALPH2O(ND2)*ZD22)**2+ZD22*ZD22
    RD11=1./D11
    RD12=1./D12

```

NRL REPORT 8074

```

RD21=1.0/D21
RD22=1.0/D22
RSRDS1=SQRT(RD11*RD21)
RSRDS2=SQRT(RD12*RD22)
HC210=0.5*HZ*(RD11+RD12)
HC220=0.5*HZ*(RD21+RD22)
HC212=0.5*HZ*(RSRDS1*RSRDS2)
WT1=C.7
WT2=HC212

```

```

C
C  COMPUTE THE PHASE CHANGE IN THIS LOOP. THEN EVALUATE
C
54  CONTINUE

```

WHITNEY, MADER, AND ULRICH

```

PHIMX=0.0
IF (NMS .EQ. 0) GO TO 80
DO 55 J=1,NY
Y=YZERO+FLOAT(J-1)*HY
DO 55 I=1,NX
X=XZERO+FLOAT(I-1)*HX
GNEW=GCON0*(WT1*EX0*EPS0(I,J)+WT2*EXN*EPS1(I,J))
PHI=0.5*PI*(HCZ10*G1(I)+HCZ20*G2(J)-GNEW)
PHI=PHI-XXX*DALPH1(ND2)-YYY*DALPH2(ND2)-A*DBE1(ND2)
B(I,J)=R(I,J)*CMPLX(COS(PHI), SIN(PHI))
IF (TENS(I,J) .LT. 0.05*TENSA) GO TO 55
PHIMX=AMAX1(PHIMX, 0.5*GNEW)
55 CONTINUE
GO TO 60

C
C COMPUTE THE PHASE CHANGE IN THIS LOOP WHEN NMS = 0
C
80 DO 85 J=1,NY
Y=YZERO+FLOAT(J-1)*HY
DO 85 I=1,NX
X=XZERO+FLOAT(I-1)*HX
GNEW=GCON0*(WT1*EX0*EPS0(I,J)+WT2*EXN*EPS1(I,J))
PHI=0.5*PI*(HCZ10*G1(I)+HCZ20*G2(J)-GNEW)
PHI=PHI-XXX*DALPH1(ND2)-YYY*DALPH2(ND2)-X*DBE1(ND2)
A(I,J)=A(I,J)*CMPLX(COS(PHI), SIN(PHI))
IF (TENS(I,J) .LT. 0.05*LENSE) GO TO 85
PHIMX=AMAX1(PHIMX, 0.5*GNEW)
85 CONTINUE
60 RETURN
END

```

NRL REPORT 8074

SUBROUTINE INTENS(A, LI)

```

C
C THIS SUBROUTINE TAPERS THE BOUNDARIES OF THE COMPUTATIONAL
C GRID TO ZERO, FINDS THE INTENSITY AT EACH GRID POINT,
C AND PLOTS THE CONTOURS OF THE FOURIER TRANSFORMED DATA.
C
C * * * * *
COMMON /AAA/ EPS(64,64),EPSD(64,64),A00(11,99),B00(19,10),
  AIN(2,64),DALPH1(2),DALPH2(2),DNE(1,2),ALPH10(2),ALPH20(2),
  BET10(2),D10(2),D20(2),KD10(2),KD20(2),SKD10(2),KGEN0(2),
COMMON /BBB/ TENS(64,64),G1(64),G2(64),PHASE1(64),PHASE2(64),
  CONMIN(10),M2(3),SV1(64),SV2(64),PARM(80)
COMMON /SINGLS/ F, PNA, PNALF, PNK, PNO, PNS, PND, PN4, MX, NY,
  HZ, Z, Z2,ZZF, ZNM, ZFINAL, XZERO, YZERO, WIDTH, ALPHA, WN,
  VODT, OMDT, HT, ENERGY, ALPHAC, CS, REFRAC, GAMMA, E1, C1K,
  EUTKJ, RMT, POUT, DAMEA, I2, IS, IPULSE, AS2,PCW31,CON1,CON2,
  Z1, RI63MX, Z2, RIMXA, Z3, APN, Z4, HZFN, DKAREX, TENSMA,
  EX, PHIMX, EPSMX, ERRMA, DGMA, R1, HDIMAX, VIERM, PHIMAX,PHIMNS,
  P1, IMAX, JMAX, NX, NY, NAD, NX2, NY2, NXY, NADIM, NYDIM, NP1,
  IPLOT,NITER,NDUF,NXN,NYI,PHS,KFLAG,D1,D2,P1,P2,ERID1,SRID2,
  RSKD12,XCEN,TLAST,SGRTA,PNDC,GCONU,GCON,DD1,DC2,10,HC220,HC21N,
  HC22N,HC212,ALPH1,ALPH2,HE1,CON1,CON2,HC2,HC3,LAG,LAGN,1,1,12
COMMON /OUTS/ NBN,SCLFAC,NK5,LP,NCK,NHEAL,NPLOT,NPUNCH
C * * * * *
COMPLEX A(64,64)
LOGICAL LI
C
C TAPER BOUNDARY VALUES TO ZERO
C
  IF (LI) GO TO 12
  DO 100 I=1,NX
    A(I,1)=0.
    A(I,NY)=0.
    A(1,2)=A(1,2)*.5
    A(1,NY-1)=A(1,NY-1)*.5
  100 DO 110 J=1,NY
    A(1,J)=0.
    A(NX,J)=0.
    A(2,J)=A(2,J)*.5
    A(NX-1,J)=A(NX-1,J)*.5
  11 CONTINUE
  120 CONTINUE
C
C COMPUTE THE INTENSITY AT EACH GRID POINT AND LOCATE THE MAXIMUM
C
  TENSXX=0.
  DO 9 J=1,NY
    DO 9 I=1,NX
      TENS(I,J)=A(I,J)*CONJG(A(I,J))
      IF (TENS(I,J) .GE. TENSXX) GO TO 9
      IMAX=I
      JMAX=J
      TENSXX=TENS(I,J)
  9 CONTINUE
C
C RETURN IF NOT PLOTTING FOURIER TRANSFORM
C
  IF (.NOT. LI) GO TO 15
C
C REPORT ARRAY WHEN PLOTTING FOURIER TRANSFORM
C
  DO 16 J=1,NY2
    DO 16 I=1,NX2
      HOLE=TENS(I,J)
      TENS(I,J)=TENS(1+NX2,J+NY2)
      TENS(1+NX2,J+NY2)=HOLE
  16 CONTINUE

```

14

WHITNEY, MADER, AND ULRICH

```

DO 16 I=1,NX2
  HOLD=TENS(I+NX2,J)
  TENS(I+NX2,J)=TENS(I,J+NY2)
  TENS(I,J+NY2)=HOLD
16  CONTINUE
C
C  PLOT FOURIER TRANSFORMS
C
  IF (NPLT.LT.5) GO TO 15
  CALL SYMBOL (0.,8.0,.14,3HZ =.0.,3)
  CALL NUMBER (1.36,8.0,.14,2 ,0.,4HF8.5)
  CALL SYMBOL (1.32,8.0,.14,4H KM ,0.,4)
  CALL LABEL(0.0,4.0)
  CALL PLOT (2.5,0.,-3)
  CALL SYMBOL (5.0, 5.0, 0.14, 3, 0, -1)
  CALL TOPOGRAPH(TENS, NXDIM, NYDIM, NX, NY, 0.0, 0.0, 11, 10.0,
    . 10.0, EPSO , XZERO, HX, 4HF6.1, 1HA, -1,YZERO,H1,4HF6.1,1H1,1)
DO 74 J=1,NY
74  EPSO(1,J)=0.0
15  RETURN

```

END

NRL REPORT 8074

```

SUBROUTINE VTRANS(A,NS)
C
C THIS SUBROUTINE CHANGES PHI BACK TO PSI (OR A) AND ADAPTS
C THE COORDINATE SYSTEM TO THE CHANGING BEAM SIZE.
C
C
C
COMMON /BBB/ TENS(64,64), G1(64), G2(64), PHASE1(64), PHASE2(64),
COMMON /AAA/ EPS1(64,64), EPS2(64,64), PARM(80)
COMMON /AAA/ EPS3(64,64), EPS4(64,64), RUS(11,11), RUS(11,11),
COMMON /AAA/ DUALPH1(2), DUALPH2(2), DPH1(1,2), ALPH1(1,2), ALPH2(1,2),
COMMON /SINGLES/ F, PNA, PNA1, PNA2, PNA3, PNA4, PNA5, PNA6, PNA7, PNA8, PNA9, PNA10, PNA11, PNA12,
COMMON /ZONES/ Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, Z12, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z20, Z21, Z22, Z23, Z24, Z25, Z26, Z27, Z28, Z29, Z30, Z31, Z32, Z33, Z34, Z35, Z36, Z37, Z38, Z39, Z40, Z41, Z42, Z43, Z44, Z45, Z46, Z47, Z48, Z49, Z50, Z51, Z52, Z53, Z54, Z55, Z56, Z57, Z58, Z59, Z60, Z61, Z62, Z63, Z64, Z65, Z66, Z67, Z68, Z69, Z70, Z71, Z72, Z73, Z74, Z75, Z76, Z77, Z78, Z79, Z80, Z81, Z82, Z83, Z84, Z85, Z86, Z87, Z88, Z89, Z90, Z91, Z92, Z93, Z94, Z95, Z96, Z97, Z98, Z99, Z100, Z101, Z102, Z103, Z104, Z105, Z106, Z107, Z108, Z109, Z110, Z111, Z112, Z113, Z114, Z115, Z116, Z117, Z118, Z119, Z120, Z121, Z122, Z123, Z124, Z125, Z126, Z127, Z128, Z129, Z130, Z131, Z132, Z133, Z134, Z135, Z136, Z137, Z138, Z139, Z140, Z141, Z142, Z143, Z144, Z145, Z146, Z147, Z148, Z149, Z150, Z151, Z152, Z153, Z154, Z155, Z156, Z157, Z158, Z159, Z160, Z161, Z162, Z163, Z164, Z165, Z166, Z167, Z168, Z169, Z170, Z171, Z172, Z173, Z174, Z175, Z176, Z177, Z178, Z179, Z180, Z181, Z182, Z183, Z184, Z185, Z186, Z187, Z188, Z189, Z190, Z191, Z192, Z193, Z194, Z195, Z196, Z197, Z198, Z199, Z200, Z201, Z202, Z203, Z204, Z205, Z206, Z207, Z208, Z209, Z210, Z211, Z212, Z213, Z214, Z215, Z216, Z217, Z218, Z219, Z220, Z221, Z222, Z223, Z224, Z225, Z226, Z227, Z228, Z229, Z230, Z231, Z232, Z233, Z234, Z235, Z236, Z237, Z238, Z239, Z240, Z241, Z242, Z243, Z244, Z245, Z246, Z247, Z248, Z249, Z250, Z251, Z252, Z253, Z254, Z255, Z256, Z257, Z258, Z259, Z260, Z261, Z262, Z263, Z264, Z265, Z266, Z267, Z268, Z269, Z270, Z271, Z272, Z273, Z274, Z275, Z276, Z277, Z278, Z279, Z280, Z281, Z282, Z283, Z284, Z285, Z286, Z287, Z288, Z289, Z290, Z291, Z292, Z293, Z294, Z295, Z296, Z297, Z298, Z299, Z300, Z301, Z302, Z303, Z304, Z305, Z306, Z307, Z308, Z309, Z310, Z311, Z312, Z313, Z314, Z315, Z316, Z317, Z318, Z319, Z320, Z321, Z322, Z323, Z324, Z325, Z326, Z327, Z328, Z329, Z330, Z331, Z332, Z333, Z334, Z335, Z336, Z337, Z338, Z339, Z340, Z341, Z342, Z343, Z344, Z345, Z346, Z347, Z348, Z349, Z350, Z351, Z352, Z353, Z354, Z355, Z356, Z357, Z358, Z359, Z360, Z361, Z362, Z363, Z364, Z365, Z366, Z367, Z368, Z369, Z370, Z371, Z372, Z373, Z374, Z375, Z376, Z377, Z378, Z379, Z380, Z381, Z382, Z383, Z384, Z385, Z386, Z387, Z388, Z389, Z390, Z391, Z392, Z393, Z394, Z395, Z396, Z397, Z398, Z399, Z400, Z401, Z402, Z403, Z404, Z405, Z406, Z407, Z408, Z409, Z410, Z411, Z412, Z413, Z414, Z415, Z416, Z417, Z418, Z419, Z420, Z421, Z422, Z423, Z424, Z425, Z426, Z427, Z428, Z429, Z430, Z431, Z432, Z433, Z434, Z435, Z436, Z437, Z438, Z439, Z440, Z441, Z442, Z443, Z444, Z445, Z446, Z447, Z448, Z449, Z450, Z451, Z452, Z453, Z454, Z455, Z456, Z457, Z458, Z459, Z460, Z461, Z462, Z463, Z464, Z465, Z466, Z467, Z468, Z469, Z470, Z471, Z472, Z473, Z474, Z475, Z476, Z477, Z478, Z479, Z480, Z481, Z482, Z483, Z484, Z485, Z486, Z487, Z488, Z489, Z490, Z491, Z492, Z493, Z494, Z495, Z496, Z497, Z498, Z499, Z500, Z501, Z502, Z503, Z504, Z505, Z506, Z507, Z508, Z509, Z510, Z511, Z512, Z513, Z514, Z515, Z516, Z517, Z518, Z519, Z520, Z521, Z522, Z523, Z524, Z525, Z526, Z527, Z528, Z529, Z530, Z531, Z532, Z533, Z534, Z535, Z536, Z537, Z538, Z539, Z540, Z541, Z542, Z543, Z544, Z545, Z546, Z547, Z548, Z549, Z550, Z551, Z552, Z553, Z554, Z555, Z556, Z557, Z558, Z559, Z560, Z561, Z562, Z563, Z564, Z565, Z566, Z567, Z568, Z569, Z570, Z571, Z572, Z573, Z574, Z575, Z576, Z577, Z578, Z579, Z580, Z581, Z582, Z583, Z584, Z585, Z586, Z587, Z588, Z589, Z590, Z591, Z592, Z593, Z594, Z595, Z596, Z597, Z598, Z599, Z600, Z601, Z602, Z603, Z604, Z605, Z606, Z607, Z608, Z609, Z610, Z611, Z612, Z613, Z614, Z615, Z616, Z617, Z618, Z619, Z620, Z621, Z622, Z623, Z624, Z625, Z626, Z627, Z628, Z629, Z630, Z631, Z632, Z633, Z634, Z635, Z636, Z637, Z638, Z639, Z640, Z641, Z642, Z643, Z644, Z645, Z646, Z647, Z648, Z649, Z650, Z651, Z652, Z653, Z654, Z655, Z656, Z657, Z658, Z659, Z660, Z661, Z662, Z663, Z664, Z665, Z666, Z667, Z668, Z669, Z670, Z671, Z672, Z673, Z674, Z675, Z676, Z677, Z678, Z679, Z680, Z681, Z682, Z683, Z684, Z685, Z686, Z687, Z688, Z689, Z690, Z691, Z692, Z693, Z694, Z695, Z696, Z697, Z698, Z699, Z700, Z701, Z702, Z703, Z704, Z705, Z706, Z707, Z708, Z709, Z710, Z711, Z712, Z713, Z714, Z715, Z716, Z717, Z718, Z719, Z720, Z721, Z722, Z723, Z724, Z725, Z726, Z727, Z728, Z729, Z730, Z731, Z732, Z733, Z734, Z735, Z736, Z737, Z738, Z739, Z740, Z741, Z742, Z743, Z744, Z745, Z746, Z747, Z748, Z749, Z750, Z751, Z752, Z753, Z754, Z755, Z756, Z757, Z758, Z759, Z760, Z761, Z762, Z763, Z764, Z765, Z766, Z767, Z768, Z769, Z770, Z771, Z772, Z773, Z774, Z775, Z776, Z777, Z778, Z779, Z780, Z781, Z782, Z783, Z784, Z785
```

WHITNEY, MADER, AND ULRICH

```

DO 15 J=1,NYM
Y=YZERO+(FLOAT(J)-0.5)*HY
DO 15 I=1,NXM
PHI=0.5*PI*(HCZ1N*G1(I+1)+HCZ2N*G2(J+1)-GCON0*
  (WT1*EX0*EPS0(I+1,J+1)+WT2*EX0*EPS(I+1,J+1)))
A(I+1,J+1)=A(I+1,J+1)*CMPLX(COS(PHI),SIN(PHI))
X=XZERO+(FLOAT(I)-0.5)*HX
AA=0.5*(A(I+1,J)+A(I,J))
AA2=CONJG(AA)*AA
SUMX=SUMX+AA2
AX=AX+AA2*X*X
BX=BX+AA2*X
FACTOR=AI*HAG(CONJG(AA)*(A(I+1,J)-A(I,J)))
CX=CX+FACTOR
DX=DX+FACTOR*X
AA=0.5*(A(I,J+1)+A(I,J))
AA2=CONJG(AA)*AA
AY=AY+AA2*Y*Y
DY=DY+AI*HAG(CONJG(AA)*(A(I,J+1)-A(I,J)))*Y
15 CONTINUE
SUMX=SUMX*DAREA
AX=AX*DAREA
BX=BX*DAREA
CX=CX*HY
DX=DX*HY
AY=AY*DAREA
DY=DY*HY
RDEFIN=1.-7*(AX*SUMX-BX*BX)
DALPH1(NS)=.5*((DX*SUMX-BX*CX)*RDEFIN)
DALPH2(NS)=.5*DY/AY
DBET1(NS)=(AX*CX-BX*DX)*RDEFIN
C
C STORE VALUES FOR LATER OUTPUT
C
K=1PLOT
AOUT(1,K)=Z
AOUT(2,K)=D
AOUT(3,K)=E1
AOUT(4,K)=E2
AOUT(5,K)=ALPH1
AOUT(6,K)=ALPH2
AOUT(7,K)=BET1
AOUT(8,K)=DALPH1(NS)
AOUT(9,K)=DALPH2(NS)
AOUT(10,K)=DBET1(NS)
AOUT(11,K)=XCEN
IF (NAD .NE. 0) GO TO 20
DALPH1(NS)=.
DALPH2(NS)=.
DBET1(NS)=.
20 CONTINUE
C
C UPDATE THE LINES AND QUANTITIES FROM THE
C
ALPH1(NS)=ALPH1+DALPH1(NS)
ALPH2(NS)=ALPH2+DALPH2(NS)
BET1(NS)=BET1+DBET1(NS)
XCEN(NS)=XCEN
C
C COMPUTE QUANTITIES TO TAKE THE LOOK-UP TABLES
C ADAPTED COORDINATES.
C
IF (NFLAG .EQ. 1) GO TO 50
P1=1.0
P2=1.0

```

NRL REPORT 8074

```

IF (NMS .EQ. 1) P2=.0
HZ1=CON1*HZ
HZ2=CON2*HZ
ZD11=HZ1*RD10(2)
ZD12=HZ2*RD10(2)
ZD21=HZ1*RD20(2)
ZD22=HZ2*RD20(2)
D11=D10(2)*((1.0+2.0*ALPH10(2)*ZD11)**2+ZD11*ZD11)
D12=D10(2)*((1.0+2.0*ALPH10(2)*ZD12)**2+ZD12*ZD12)
D21=D20(2)*((1.0+2.0*ALPH20(2)*ZD21)**2+ZD21*ZD21)
D22=D20(2)*((1.0+2.0*ALPH20(2)*ZD22)**2+ZD22*ZD22)
RD11=1.0/D11
RD12=1.0/D12
RD21=1.0/D21
RD22=1.0/D22
R5RDS1=SGRT(RD11*RD21)
R5RDS2=SGRT(RD12*RD22)
W11=P2*(HCZ12-HZ1*R5RDS1-HZ2*R5RDS2)
W12=HCZ12-W11
NFLAG=1
RETURN
END

```

50

WHITNEY, MADER, AND ULRICH

SUBROUTINE OUTPUT(LP)

THIS SUBROUTINE STORES OUTPUT DATA DURING THE CALCULATIONS AND PRINTS, PLOTS AND PUNCHES VARIOUS OUTPUT DATA WHEN THE CALCULATIONS HAVE CONCLUDED.

```

C * * * * *
COMMON /AAA/ EPS(64,64),EPSO(64,64),AOUT(11,99),BOOT(9,10),
  . AIN(2,64),DALPHI(2),DALPH2(2),DBE1(12),ALPH10(2),ALPH20(2),
  . BE11(2), D10(2), D20(2), RD10(2), RD20(2), SRD10(2), XCEMS(2),
COMMON /999/ A1(64,64), A2(64,64), ITENS(64,64)
COMMON /RRM/ TENB(64,64), G1(64), G2(64), PGRM1(64), PGRM2(64),
  . CONMINT(1),SV2(3),SV1(64),SV2(64),PARA(1,8)
COMMON /51/NGLS, F, PNA, PHAL, PAK, PNO, RNS, PND, PNZ, IX, MY,
  . HZ, Z, ZF,ZZF, ZNM, ZFINAL, XZERO, YZERO, WIDTH, ALPH, SN,
  . V, BT, BOBT, HT, ENRGY, ALPHAC, CS, KERRAC, GAMMA, LTJ, CTK,
  . EUTKJ, RHT, POUT, DAKA, Z, TS, TPULSE, AGZ,PCR,S1,TCOKI,ICORZ,
  . Z1, R163HX, Z2, R163HX, Z3, ARVN, Z4, HZEN, DKARL, T1NSMX,
  . EX, PH1PX, LP5HX, ERMX, GOMX, K1, RDINAX, VIRM, PH1MX,HZNNS,
  . P1, IMAX, JMAX, NX, NY, M4, NX2, NY2, NXY, NXDM, HYDM, NPT,
  . IPLOT, E1TR,NDIB, NX, NY, S1, S1FAC, D1, D2, D1F, ZKST, D1SRD2,
  . RSRD12,XCEN,TLAST,SRD1B,PHI,CCN,CCN1,CCN2,10,CC220,CC219,
  . CC22N,CC212,ALPH1,ALPH2,CC1,CC2,CC3,CC4,CC5,CC6,CC7,CC8,CC9,CC10,CC11,CC12
COMMON /OUT5/ A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB,AC,AD,AE,AF,AG,AH,AI,
COMMON /PUB/ND

```

14

[illegible]

```

1 PLOT=1 PLOT+1
1 PLOT=MINT(1 PLOT,8.)
PTRANS=POUT*FX
1 TRANS=ENERGY*FX
SCFAC1=WIDTH*SRTO1
SCFAC2=WIDTH*SRTO2
SCAREA=SCFAC1*SCFAC2
R1PEAK=1.:ZTINSMX

```

COMPUTE THE AREA AND FLUX CONTAINED WITHIN EACH CONTOUR LEVEL.

```

      SUMA=0
      DO 46 I=1,10
      SUPA(I)=AREA(I)*
46      DO 52 J=1,NX
      DO 52 I=1,NX
          SUM=SUM+TENS(I,J)
          FRAC=TENS(I,J)*RIPKAK
      DO 50 N=1,10
          IF (FRAC .LT. COMBINT( I )) GO TO 54
      DO 48 L=N,10
          SUMA(L)=SUPA(L)+TENS(I,J)
48      AREA(L)=AREA(L)+OAPLX
      GO TO 52
50      CONTINUE
52      CONTINUE

```

CORRECT UNITS AND NORMALIZE AREA AND FLUXES

SUBP=1,1415A21 A
 100 54 121,110

NRL REPORT 8074

```

APRINT(1)=AREA(1)*SCAREA
AF(1)=AREA(1)/AREA(9)
FF(1)=SUMA(1)*DAREA
FLUX(1)=FF(1)*PTRANS
RI(1)=FLUX(1)/APRINT(1)
54  CONTINUE
C
C  INTERPOLATE TO FIND THE AREA CONTAINING 0.63 OF TOTAL POWER
C
      DO 661 I=1,10
      IF(FF(I),LE,0.63) GO TO 661
      ILO=I-1
      IHI=I
      GO TO 662
661  CONTINUE
C  IF 0.63 EXCEEDS RANGE OF FF, USE LAST TWO FF VALUES TO EXTRAPOLATE
      ILO=9
      IHI=10
662  AP=APRINT(ILO)+((0.63-FF(ILO))/(FF(IHI)-FF(ILO)))*
      1  (APRINT(IHI)-APRINT(ILO))
C
C  FIND AREA RELATIVE TO INFINITE GAUSSIAN BEAM
C
      AP1=PI*W2*D
      A REL=AP/AP1
      RI63=0.63*PTRANS/AP
      RIMX=TENSMX*PTRANS/SCARLA
C
C  COMPUTE OTHER QUANTITIES
C
      ZCOR(IPLOT)=Z
      AVI(IPLOT)=0.63*PTRANS/AP
      A63(IPLOT)=AP
      XMI(IPLOT)=RIMX
      XNX(IPLOT)=(I*MAX-NX2)*SCFAC1*HX
      XMY(IPLOT)=(J*MAX-NY2)*SCFAC2*HY
      HZ2(IPLOT)=HZ
      EPSMAX(IPLOT)=EPSMX
      PHIMAX(IPLOT)=PHIMX
      PARM(IPLOT)=1.0/VTERR
      IF (AVI(IPLOT) .GT. RI63MX) Z1=Z
      RI63MX=AMAX1(AVI(IPLOT), RI63MX)
      IF (XMI(IPLOT) .GT. RIMXX) Z2=Z
      RIMXX=AMAX1(XMI(IPLOT), RIMXX)
      IF (A63(IPLOT) .LT. APMN) Z2=Z
      APMN=AMIN1(A63(IPLOT), APMN)
      IF (Z .GE. ZFINAL) GO TO 241
      IF (HZ .LT. HZMN) Z4=Z
      HZMN=AMIN1(HZ, HZMN)
C
241  IF (.NOT. LP) GO TO 77
C
C  PRINT RESULTS
C
      IF (NEXIT.NE.0) PRINT 149
149  FORMAT(///26X2PH*** PREATURE EXIT ***//)
      REL=KIMX*AP1/PTRANS
      TENSMX=TENSMX*DAREA
C
      PRINT 149*
149  FORMAT(/// 44X15H*** RESULTS ***//)
      PRINT 15 , Z,500,NITER,IPOOT,PTRANS,AP,AK L,0.63,0.1 X,0.01
150  FORMAT(24X2BHIDE CALCULATIONS REACHED Z =E10.5,1X0.0001//
      X 17X35HIDE SUM OVER PARALIZED INTENSITY =E10.5,//
      X 29X2PHIDE NUMBER OF Z-STEPS =15,//

```

WHITNEY, MADER, AND ULRICH

```

X 12X40HAVERAGE POWER (KW) EMITTED AT APERTURE =F10.3, //
X 20X32HAVERAGE TRANSMITTED POWER (KW) =F10.3, //
X 14X38HAREA (SUCM) CONTAINING 0.63 OF POWER =F10.3, //
X 17X35HAREL (RELATIVE TO INF. GAUSSIAN) =F10.3, //
X 10X42HAVERAGE INTENSITY (KW/SUCM) IN THIS AREA =F10.3, //
X 26X26HPPAK INTENSITY (KW/SUCM) =F10.3, //
X 12X40HREL (RELATIVE TO INF. GAUSSIAN PLAK) =F10.5)

C
C PRINT NUMERICAL DATA
C
PRINT 430,
430 FORMAT(1H15CX22H*** NUMERICAL DATA ***//)
PRINT 160,
160 FORMAT( /
, 9X *RANGE(K)* 5X *AS2(CM*2)* 6X *AS2/A2D* 6X *TP(SEC)* 7X
, *PCR(W)* 4X *TSAT(SEC)* 2X *ISAT(G/CM2)* 5X *TURBCOR1* 5X
, *TURBCOR2*)
DO 170 I=1,10
170 PRINT 180, (ROUT(K,I),K=1,9)
180 FORMAT(5X,3F13.5,6E13.4)
PRINT 420,
420 FORMAT(17X40HNORMALIZED AMPLITUDE SAMPLES AT APERTURE//)
IPX=NX/2
IPY=NY/2
PRINT 400, IPX, NY, (AIN(1,K),K=1,NY)
400 FORMAT(5X5HAT X=13,2X4HY= 12X2HT014, //2(32F4.1, //))
PRINT 410, IPY, NX, (AIN(2,K),K=1,NX)
410 FORMAT(5X5HAT Y=13,2X4HX= 12X2HT014, //2(32F4.1, //))
C
PRINT 170,
170 FORMAT(//5X1HZ7X2HZ2X1HD6X2HD15X2HD25X6HALPHA12X6HALPHA2
1 3X5HBETA12X6HALPHA12X6HALPHA23X5HDELTA16X4HXC1NXX5HLEP5MX
2 3X5HPHIMX4X4HPARM//)
JPLUT=1PLOT=1
DO 130 K=1,JPLUT
130 PRINT 140, (ROUT(1,K),HZZ(K), (ROUT(1,K),I=2,11),
1 EPSMAX(K),PHIMAX(K),PARM(K))
140 FORMAT(1XF6.3,(E8.1,3F7.4, //E8.0,EF.1,F8.4,EB.1)
C
C PRINT OUTPUT DATA
C
PRINT 220,
220 FORMAT(1H15CX19H*** OUTPUT DATA ***//)
PRINT 62,
62 FORMAT( /,37X*AREA*8X*FLUX*6X*AREA*6X*FLUX*5X*IRRADIANCL*
, /,26X *LEVEL* 4X
1 *(5X CM)*7X*(KW)*4X*FRACTION*2X*FRACTION*3X*(KW/SQ-CM)*)
DO 66 I=1,10
PRINT 64, (CONFIN(I),APERT(I),FLUX(I),AF(I),EF(I),EB(I))
64 FORMAT(25X,F6.4,6E12.3,6F9.4,F10.4,1E14.4)
66 CONTINUE
PRINT 70, R163MX,Z1,R1XX,Z2,AP14,Z3,HZ5.4,Z4
70 FORMAT(//40X* MAXIMUM AVG 1 **
7 E10.3,2X,*AT Z=*,E10.3,/,40X,* MAXIMUM PEAK 1=*,E10.3,2X,*AT Z=*,
8 E10.3,/,40X,* MINIMUM AREA ***,E10.3,2X,*AT Z=*,E10.3,/,40X,
9 * MINIMUM HZ ***,E10.3,2X,*AT Z=*,E10.3)
PRINT 190
190 FORMAT(//31X1HZ9X4HIAVE 7X3HA636X4HIMAX5X5HXPLAK5X5HYPEAK)
DO 200 K=1,1PLOT
200 PRINT 210, ZCOR(K),AVI(K),A63(K),XNI(K),XMX(K),XMY(K)
210 FORMAT(25X6F10.3)
C
C PUNCHED OUTPUT
C
IF (NPUNCH.EQ.0) GO TO 230
PW10=WIDTH

```

BEST AVAILABLE COPY

53

WHITNEY, MADER, AND ULRICH

54

NRL REPORT 8074

```

SUBROUTINE LABEL(X,Y)
COMMON /SINGLS/ F, PNA, PNALE, PNA, PNO, PNS, PND, PNZ, PX, PY,
  HZ, Z, ZZ, ZZF, ZNM, ZFINAL, XZERO, YZERO, WIDTH, ALPHA, WN,
  VODT, OMDT, HT, ENERGY, ALPHAC, GS, KIFRAC, GAMMA, LTJ, CTK,
  EUTKJ, RMT, POUT, DAREA, W2, TS, TPULSE, AS2, PCN, S1, TCOR1, TCOR2,
  Z1, H163MX, Z2, R1MXMX, Z3, APMN, Z4, HZIN, DKAREA, TENSIX,
  EX, PHIMX, EPSMX, ERMX, DGMX, R1, HUIMAX, VTERM, PHIMX, HZNM,
  PI, IMAX, JMAX, NX, NY, MAD, SX2, NY2, NXY, NXDIN, NYDIN, NPT,
  IPLOT, NITER, NHUF, NXI, NYN, NMS, NFLAG, D1, D2, P1, P2, SRTD1, SKTD2,
  RSKD12, XCEN, TLAST, SORTB, PND, GCON, GCON, DOL, HCZ10, HCZ20, HCZ14,
  HCZ24, HCZ12, ALPH1, ALPH2, DELT1, CON1, CON2, HZC, HZL, LXC, EXN, D11, WT2
COMMON /OUTS/ NBM, SCLFAC, NRS, NPM, NCW, NEXIT, NPLOT, NPUNCH
X1=X+.75
IF (NPM,LT,0) GO TO 1

```

14

```

C
CALL SYMBOL(X,Y,.1,50,ALF=.0,7.5)
CALL NUMBER(X1,Y,.1,PNALE=.0,7.3)
C
Y=Y+.2
CALL SYMBOL(X,Y,.1,50, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,PNA,7.3)
C
Y=Y+.2
CALL SYMBOL(X,Y,.1,50, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,PNA,7.3)
C
Y=Y+.2
CALL SYMBOL(X,Y,.1,50, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,PNA,7.3)
C
Y=Y+.2
CALL SYMBOL(X,Y,.1,50, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,PNA,7.3)
C
RETURN
CONTINUE
P=1/1
CALL SYMBOL(X,Y,.1,70, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,P,7.2)
C
Y=Y+.2
CALL SYMBOL(X,Y,.1,70, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,P,7.2)
C
Y=Y+.2
ALF=ALPHA*.1
CALL SYMBOL(X,Y,.1,70, ALF=.0,7.5)
CALL NUMBER(X1,Y,.1,ALF,7.2)
C
Y=Y+.2
CALL SYMBOL(X,Y,.1,70, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,P,7.1)
C
Y=Y+.2
V=V+1/1/1/1
CALL SYMBOL(X,Y,.1,70, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,P,7.1)
C
Y=Y+.2
CONTINUE
CALL SYMBOL(X,Y,.1,70, NK=.0,7.5)
CALL NUMBER(X1,Y,.1,P,7.1)

```

WHITNEY, MADER, AND ULRICH

```
C      Y=Y+0.2
      CALL SYMBOL(X,Y,0.1,7H      DT=0.0,7)
      CALL NUMBER(X1,Y,0.1,HT=0.0,4HF7.5)

C      Y=Y+0.2
      CALL SYMBOL(X,Y,0.1,7HENERGY=0.0,7)
      CALL NUMBER(X1,Y,0.1,ENERGY,0.0,4HF9.1)
      RETURN
      END
```

NRL REPORT 8074

BEST AVAILABLE COPY

BEST COPY AVAILABLE

WHITNEY, MADER, AND ULRICH

```

SUBROUTINE KONTUR (F,NXDIM,NYDIM,NX,NY,FLVEL,XINCHES,YINCHES,
1 IMAGE)
  DIMENSION F(NXDIM,NYDIM),IMAGE(NXDIM,NYDIM)

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

  XFACTOR = XINCHES/(NX-1)
  YFACTOR = YINCHES/(NY-1)
  XD=XINCHES/2.
  YD=YINCHES/2.

C   LOAD IMAGE ARRAY

  DO 2 IY=1,NY
  DO 2 IX=1,NX
    IF (F(IX,IY).GE.FLVEL) 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
  CONTINUE
4  RETURN

C   LIFT PEN AND BRING TO STARTING POINTS AND PRINT THE REGION NUMBER.

5  IYSTART = IY
  IF (IY.EQ.1) GO TO 6
  IF (IMAGE(IX,IY-1).EQ.0) GO TO 8
  CYO = (IY-1)-(F(IX,IY)-FLVEL)/Z0(IX,IY)-(F(IX,IY-1))*YFACTOR
  GO TO 7
6  CYO = 0
7  CXO = (IX-1)*XFACTOR
  CXOP=CXO+XD
  CYOP=CYO+YD
  CALL PLOT(CXOP,CYOP,3)
  INOUT = 2
  GO TO 20

C   START AN INNER BOUNDARY
  INOUT = 1
  CXO = (IX-2)*XFACTOR
  CYO=(IY-2)+(F(IX-1,IY-1)-FLVEL)/Z0(IX-1,IY-1)-(F(IX-1,IY))*YFACTOR
  CXOP=CXO+XD
  CYOP=CYO+YD
  CALL PLOT(CXOP,CYOP,3)
  GO TO 20

C   SKIRT DIRECTION IS ALWAYS CLOCKWISE FOR AN EXTERNAL BOUNDARY,
C   AND COUNTER-CLOCKWISE FOR AN INTERNAL BOUNDARY.
C   (THE INSIDE OF THE REGION IS ALWAYS TO THE RIGHT OF THE SKIRT DIRECTION)

C   POSITIVE X-CROSSING

10  CX = (IX-1)*XFACTOR
  IF (IY.EQ.NY) GO TO 11
  CY = (IY-1)+(F(IX,IY)-FLVEL)/Z0(IX,IY)-(F(IX,IY+1))*YFACTOR

```

NRL REPORT 8074

```

GO TO 12
11 CY = (NY-1)*YFACTOR
12 CONTINUE
   CXP=CX-XD
   CYP=CY-YD
   CALL PLOT(CXP,CYP,2)

   IF (CX.NE.CXD) GO TO 16
   YGAP = ARSF(CY - CYD)
   IF (YGAP.LT..001) GO TO 3

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

C POSITIVE Y-CROSSING
20 IF (IX.EQ.1) GO TO 21
   CX = (IX-1-(F(IX,1Y) - FLEVEL)/(F(IX,1Y) - F(IX-1,1Y)))*XFACTOR
   GO TO 22
21 CX = 0
22 CY = (1Y-1)*YFACTOR
   CXP=CX-XD
   CYP=CY-YD
   CALL PLOT(CXP,CYP,2)

   DO 26 I=IX,NX
   IF (IPAGE(1,1Y).LT.1) GO TO 25
26 IPAGE(1,1Y) =

25 IF (1Y.EQ.NY) GO TO 1
   IF (F(IX,1Y+1) - FLEVEL) 10,23,23
23 IF (1X.EQ.1) GO TO 24
   IF (F(1X-1,1Y+1) - FLEVEL) 24,25,25
24 1Y = 1Y + 1
   GO TO 2
25 1X = 1X-1
   1Y = 1Y+1
   GO TO 30

C NEGATIVE X-CROSSING
30 CX = (1X-1)*XFACTOR
   IF (1Y.EQ.1) GO TO 31
   CY = (1Y-1-(F(1X,1Y) - FLEVEL)/(F(1X,1Y) - F(1X,1Y-1)))*YFACTOR
   GO TO 32
31 CY = 0
32 CONTINUE
   CXP=CX-XD
   CYP=CY-YD
   CALL PLOT(CXP,CYP,2)

   IF (CX.NE.CXD) GO TO 33
   IF (CY.NE.CYD) GO TO 3

33 IF (1X.EQ.1) GO TO 25
   IF (F(1X-1,1Y) - FLEVEL) 20,26,26
36 IF (1Y.EQ.1) GO TO 35
   IF (F(1X-1,1Y-1) - FLEVEL) 35,36,36
35 1X = 1X-1

```

WHITNEY, MADER, AND ULRICH

```

      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+(F(IX,IY) - FLEVEL)/(F(IX,IY) - F(IX+1,IY)))*XFACTOR
    GO TO 42
41  CX = (NX-1)*XFACTOR

42  CONTINUE
    CXP=CX-XD
    CYP=CY-YD
    CALL PLOT(CXP,CYP,2)

    IF (IY.EQ.1) GO TO 30
    IF (F(IX,IY-1) - FLEVEL) 30,43,43
43  IF (IX.EQ.NX) GO TO 44
    IF (F(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

```

NRL REPORT 8074

```

SUBROUTINE CONTOUR (F,NXDIM,NYDIM,NX,NY,FLEVEL,XINCHES,YINCHES,
1  IMAGE)
  DIMENSION F(NXDIM,NYDIM),IMAGE(NXDIM,NYDIM)

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.

  XFACTOR = XINCHES/(NX-1)
  YFACTOR = YINCHES/(NY-1)

C   LOAD IMAGE ARRAY

  DO 2 IY=1,NY
  DO 2 IX=1,NX
    IF (F(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
  DO 4 IY=IYSTART,NY
  DO 4 IX=1,NX
    IF (IMAGE(IX,IY).EQ.1) GO TO 5
  CONTINUE
  RETURN

C   LIFT PEN AND GOING TO STARTING POINT, AND GOING TO REGION FOUND.

  IYSTART = IY
  IF (IY.EQ.1) GO TO 6
  IF (IMAGE(IX,IY-1).EQ.0) GO TO 6
  CY = (IY-1-(F(IX,IY)-FLEVEL)/F(IX,IY)-F(IX,IY-1))*YFACTOR
  GO TO 7
6  CY = 0
7  CX = (IX-1)*XFACTOR
  CALL PLOT (CX,CY,0)
  INOUT = 2
  GO TO 2

C   START AN INTERIOR BOUNDARY

  INOUT = 1
  CX = (IX-2)*XFACTOR
  CY = (IY-2+(F(IX-1,IY-1)-FLEVEL)/F(IX-1,IY-1)-F(IX-1,IY))*YFACTOR
  CALL PLOT (CX,CY,0)
  GO TO 20

C   SPLIT DIRECTION IS ALWAYS CLOCKWISE FOR AN EXTERNAL BOUNDARY,
C   AND COUNTER-CLOCKWISE FOR AN INTERNAL BOUNDARY.
C   THE DIRECTION OF THE REGION IS ALWAYS TO THE RIGHT OF THE FIRST DIRECTION.

C   POSITIVE X-CROSSING

10  CX = (IX-1)*XFACTOR
  IF (IY.EQ.NY) GO TO 11
  CY = (IY-1+(F(IX,IY)-FLEVEL)/F(IX,IY)-F(IX,IY-1))*YFACTOR
  GO TO 12
12  CALL PLOT (CX,CY,0)
11  CY = (NY-1)*YFACTOR

  IF (CX.EQ.CX-1) GO TO 16
  YCAP = ABS(CY - CY0)
  IF (YCAP.EQ.0) GO TO 14

```

BEST AVAILABLE COPY

WHITNEY, MADER, AND ULRICH

BEST AVAILABLE COPY

```

16 IF (IX.EQ.NX) GO TO 40
   IF (F(IX+1,IY) = FLEVEL) 40,13,13
13 IF (IY.EQ.NY) GO TO 14
   IF (F(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)*(F(IX,IY) - FLEVEL)/(F(IX,IY) - F(IX-1,IY))*XFACTOR
   GO TO 22
21 CX = 0
22 CY = (IY-1)*YFACTOR
   CALL PLOT(CX,CY,2)

   DO 26 I=IX,NX
   IF (IMAGE(I,IY).LT.1) GO TO 20
26 IMAGE(I,IY) = 0

28 IF (IY.EQ.NY) GO TO 10
   IF (F(IX,IY+1) = FLEVEL) 10,23,23
23 IF (IX.EQ.1) GO TO 24
   IF (F(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 Y-CROSSING

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

   IF (CX.EQ.CX) GO TO 33
   IF (CY.EQ.CY) GO TO 34

34 IF (IX.EQ.1) GO TO 20
   IF (F(IX-1,IY) = FLEVEL) 20,34,34
36 IF (IY.EQ.1) GO TO 35
   IF (F(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)*(F(IX,IY) - FLEVEL)/(F(IX,IY) - F(IX+1,IY))*XFACTOR
   GO TO 42
41 CX = (NX-1)*XFACTOR
42 CALL PLOT(CX,CY,2)

```

NRL REPORT 8074

```
IF (IY.EQ.1) GO TO 30
IF (F(IX,IY-1) - FLEVEL) 30,43,43
43 IF (IX.EQ.NX) GO TO 44
IF (F(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
```

WHITNEY, MADER, AND ULRICH

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 = ABS(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./D
RETURN

6 DY = 2.*10.**IEXP
TICK = 2./D
RETURN

7 PRINT 1.0, DY
100 FORMAT (13H *****ERROR IN GETTICK -- DY = 110.37)
RETURN
END
SUBROUTINE NORMAL (ARG, IEXP)

C NORMAL TAKES ANY NUMBER, ARG, AND NORMALIZES IT, I.E. 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
1 SIGN = -1.0
ARG = -ARG
2 IF (ARG - 10.0) 2,4,6
3 IF (ARG - 1.0) 3,5,5
4 ARG = ARG*10.0
IEXP = IEXP + 1
GO TO 1
5 ARG = ARG/10.0
IEXP = IEXP - 1
GO TO 1
6 ARG = SIGN*ARG
RETURN
END

BEST AVAILABLE COPY

NRL REPORT 8074

```

SUBROUTINE GETSCALE (Y,N,HEIGHT,YMIN,DY,K,TICK,IFORMAT)

C GETSCALE OBTAINS SCALING PARAMETERS FOR THE N VALUES IN ARRAY Y
C (OF DIMENSION N*K). HEIGHT IS THE GRAPH HEIGHT IN INCHES.
C YMIN,DY,TICK, AND IFORMAT ARE PROVIDED BY GETSCALE 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 ENTRY GETSCALZ INCLUDES ZERO AMONG THE VALUES SCALED
C DATA IN ARRAY Y MAY BE SCALED BY THE FOLLOWING CONVERSION TO INCHES
C Y(I) = (Y(I) - YMIN) / DY * TICK
C ENTRY DOSCALE AND DOSCALEZ CHANGE THE VALUES IN ARRAY Y TO INCHES
C ENTRY SCALE REPLACES THE SYSTEM ROUTINE SCALE (USING IFORMAT PARAMETER)

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

ENTRY GETSCALZ
IGATE=1
YMIN = YMAX = Y(1)
GO TO 1
ENTRY DOSCALE
IGATE=2
YMIN=YMAX=Y(1)
GO TO 1

ENTRY DOSCALEZ
IGATE=2
YMIN=YMAX=0
GO TO 1

ENTRY SCALE
IGATE=3
YMIN=YMAX=Y(1)

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 HEIGHT INCHES
6 DY = TICK = 1.0
IF (YMIN.EQ.0.) GO TO 10
CALL NORMAL (YMIN,1EXP)
TMIN=INT(YMIN)
IF (TMIN.GT.YMIN) TMIN=TMIN-1.
YMIN=TMIN*10.**1EXP
DYINCH=DY*(1.**(1EXP+1))/HEIGHT
CALL GETTICK (DY,TICK)
GO TO 8

C ROUND DOWN YMIN TO DY SCALE
7 DY = (YMAX - YMIN) / HEIGHT
CALL GETTICK (DY,TICK)
D=DY
CALL NORMAL (D,1DY)

```

BEST AVAILABLE COPY

```

CALL NORMAL (YMIN, IY)
X=YMIN*10.** (IY-IDY)
TX=INTF(X/D)*D
IF (TX.GT.X) TX=TX-D
YMIN=TX*10.**IDY

C   ROUND YMAX UP TO DY SCALE (SO MAX Y VALUE FALL IN LAST TICK SPACE)
CALL NORMAL (YMAX, IX)
X=YMAX*10.** (IX-IDY)
TX=INTF(X/D)*D
IF (TX.LT.X) TX=TX+D
YMAX=TX*10.**IDY

C   ADJUST TICK LENGTH FOR WIDER RANGE AND ROUNDOFF
DYINCH=(YMAX-YMIN)/HEIGHT
TICK=DY/DYINCH =.001 TO .01
IF (YMIN.LT.0.) YMIN=INTF(YMIN/DY)*DY

8  CONTINUE
IF (IGATE.EQ.1) GO TO 10
DO 9 I=1,M,K
9  Y(I)=(Y(I)-YMIN)/DYINCH
IF (IGATE.EQ.3) RETURN

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.3) IRANGE=IRANGE+NEXP

C   CONSTRUCT FORMAT
13 IDEC=IDEC*8**8
IRANGE=IRANGE*8**12
IFORNT=4HF7.3,OR,1DEC,OR,IRANGE
RETURN
14 IFORNT=4HE8.1
RETURN
END

```

NRL REPORT 8074

SUBROUTINE MULTIPLY (X,Y,N,NYS,KX,KY,KYS,XLENGTH,THEIGH,LABEL,NA
1,YLABEL,NYC,LINELABL,MARK,NMARK,XMINVAL,DY,YMINVAL,DY,TICK)

C
C MULTIPLY WILL PLOT NYS LINES OF N POINTS EACH, ON THE SAME GRAPH.
C ON A LINEAR OR LOG SCALE, DEPENDING ON THE ENR. USED.
C X VALUES ARE TAKEN FROM ARRA: A A(1), A(1+KX),...,A(1+(N-1)KX)
C THE 1ST Y ARRAY IS STORED IN Y(1), Y(1+KY),...,Y(1+(N-1)KY)
C SUCCESSIVE Y ARRAYS BEGIN KYS LOCATIONS APART.
C THE PLOT WILL BE XLENGTH INCHES LONG BY THEIGH INCHES HIGH.
C EG. THE 2ND ARRAY BEGINS AT Y(1+KYS), THE LAST AT Y(1+NYS-1)KYS)
C THE X-AXIS WILL BE LABELED WITH THE N CHARACTERS IN LABEL, AND
C MAY BE HOLLERITH FORMATS, EG. SHENERG, OR LOCATIONS WHERE THE
C CHARACTER CODES ARE STORED.)
C LINELABL IS AN ARRAY OF NYS HOLLERITH WORDS WHICH WILL BE USED TO
C IDENTIFY THE NYS LINES WHENEVER NYS.GT.1
C MARK IS AN ARRAY OF NYS INTEGERS DENOTING SPECIAL SYMBOLS (EG. -1=NOTHING,
C 0=SQUARES, 1=OCTAGON,...) TO BE DRAWN ON EACH LINE EVERY NMARK POINTS.
C A NEGATIVE NMARK SUPPRESSES THE CONNECTING LINE BETWEEN SYMBOLS.

C
C ENTRY LOGPLOT WILL GIVE A LOG-LOG PLOT
C ENTRY SEMILOGX WILL GIVE A SEMILOG PLOT WITH X ON THE LOG SCALE.
C ENTRY SEMILOGY GIVES A SEMILOG PLOT WITH Y ON THE LOG SCALE.
C ENTRY LINEPLOT GIVES A LINEAR PLOT.

C
C OUTPUT PARAMETERS XMINVAL, DY, YMINVAL, DY, GIVE INFORMATION CONCERNING
C THE SCALING OF THE PLOT, SO THAT A PT. P(X,Y) MAY BE PLOTTED ON THE
C SAME GRAPH BY USING THE FOLLOWING CONVERSION TO INCHES--
C X(INCHES) = LOG(PX/XMINVAL)*DX
C Y(INCHES) = LOG(PY/YMINVAL)*DY FOR LOGPLOT.
C WITH A LINEAR OR SEMILOG PLOT, SUBSTITUTE ONE OR BOTH OF THE FOLLOWING--
C Y(INCHES) = (PY - YMINVAL)*DY FOR LINEPLOT AND SEMILOGX
C X(INCHES) = (PX - XMINVAL)*DX FOR LINEPLOT AND SEMILOGY

C
C NB IT IS EXPECTED THAT THE CALLING PROGRAM WILL HAVE ALREADY MADE
C THE INITIAL PLOTS CALL, AND WILL ALSO CALL STOPPLOT TO TERMINATE PLOTTING.
C ALLPLOTS INCORPORATES THE PROCEDURES OF SCALE,LINE,AXIS, AND/OR LOGPLOTS,
C WITHOUT CHANGING THE CONTENTS OF THE ARRAYS PLOTTED.
C ALLPLOTS MUST BE USED WITH PLAIN (GG) PLOTTER PAPER, AS IT PRODUCES
C A VARIABLE TICK AND/OR LOGCYCLE LENGTH. (SCALING ROUTINES SUITABLE FOR
C USE WITH AVAILABLE LINED PLOTTING PAPER ARE LOGPAPER AND SCALELEN.)
C ADDRESS ANY QUESTIONS TO JEANNE GLRICH, EXT. 2326.

C
C DIMENSION X(1), Y(1), LINELABL(1)
C DATA (EXPSIZE=.07), (TENSZIE=.1), (IZERO=0), (H=.07)

C
C ENTRY LOGPLOT
C IGATE=1
C GO TO 1

C
C ENTRY SEMILOGY
C IGATE=3
C GO TO 19

C
C ENTRY SEMILOGX
C IGATE=2
C GO TO 1

C
C ENTRY LINEPLOT
C IGATE=4
C GO TO 19

C
C LOG-SCALING OF X
C 1ST FIND XMIN AND XMAX VALUES

1 XMIN = X(1)
XMAX = X(1)

WHITNEY, MADER, AND ULRICH

```

M = N*KX
DO 5 I=2,M,KX
IF (X(I) - XMIN) 2,5,3
2 XMIN = X(I)
GO TO 5
3 IF (XMAX - X(I)) 4,5,5
4 XMAX = X(I)
5 CONTINUE
IF (XMIN) 11,11,12
11 IGATE=IGATE+2
GO TO 19
12 CONTINUE

C
C DETERMINE APPROPRIATE SCALING
C VALUES ON THE GRAPH RANGE FROM XMINVAL (AT 0 INCHES) TO 10**MAXEXP
C (AT XLENGTH INCHES).
C NSCALES = NUMBER OF SCALES (POWERS OF TEN) FOR THIS DATA SET.
C FIPS = INCHES / SCALE
C
CALL NORMAL(XMIN,IBASEXP)
XMINVAL = 10.0**IBASEXP
XMINLOG = LOG(XMINVAL)
ARG = XMAX
CALL NORMAL(ARG,MAXEXP)
IF (ARG.EQ.1.0) GO TO 6
MAXEXP = MAXEXP + 1
6 NSCALES = MAXEXP - IBASEXP
FIPS = XLENGTH/NSCALES
IF (FIPS.GT.0.5) GO TO 7
XLENGTH = NSCALES
FIPS = 1
7 DX = FIPS/LOG(10.0)

C
C TO SCALE XARRAY TO INCHES
C XMINVAL CORRESPONDS TO 0 INCHES; XMINLOG = LOG(XMINVAL)
C DX = INCHES / DATA-LOG-UNITS ; IF DX CONVERTS LOG-UNITS TO INCHES
C LOG(10) = ONE SCALE IN DATA-LOG-UNITS
C LOG(10)*DX = FIPS = ONE SCALE IN INCHES
C
DRAW AND LABEL X-AXIS

YDN = -.08
YDDN = -.2
EXPLOCY = -.30
TENLOCY = -.40
CALL PLOT(0,0,3)
IEXP = IBASEXP
DO 10 I=1,ZERO,NSCALES
XLOC = XLOC + I*FIPS
CALL PLOT(XLOC,0,2)
CALL PLOT(XLOC,YDDN,2)
TENLOC = XLOC+.15
CALL SYMBOL(TENLOC,TENLOCY,1,1,SIZE,2,HI,0,0,2)
CALL NUMBER(XLOC,EXPLOCY,EXP,SIZE,1,EXP,0,2,HI,3)
CALL PLOT(XLOC,0,3)
IF (I.EQ.NSCALES) GO TO 11
IEXP = IEXP + 1
DO 9 J=2,9
XLOC = XLOC + LOG(FLOAT(J)*DX)
CALL PLOT(XLOC,0,2)
CALL PLOT(XLOC,YDN,2)
IF (FIPS.LT.1.0) GO TO 9
CALL NUMBER(XLOC,YDDN,EXP,SIZE,1,EXP,0,2,HI,1)
9 CALL PLOT(XLOC,0,3)
10 CONTINUE
XLABEL = XLENGTH + .5

```

NRL REPORT 8074

```

CALL SYMBOL(XLABLOC,0,TENSIZE,XLABEL,0,NX)
GO TO (20,38), IGATE

C
C
19 FOR NON-LOG SCALING OF X
CALL GETSCALE (X,N,XLENGTH,XMIN,DX,KX,XIICK,XFORM)
CALL AXIS (0,0,XLABEL,-NX,XLENGTH,0,XIICK,XMIN,DX,XFORM)
GO TO (20,20,20,38), IGATE

C
C
LOG-SCALING OF Y'S
20 YLENGTH=YHEIGHT
IF (YHEIGHT.GT.10.) YLENGTH=10.
21 YMIN = Y(1)
YMAX = Y(1)
MYS=NY5*KYS
DO 25 I=1,MYS,KYS
M=1+K*Y-1
DO 25 I=1,M,KY
IF (Y(I) -YMIN) 22,25,23
22 YMIN = Y(I)
GO TO 25
23 IF (YMAX - Y(I)) 24,25,25
24 YMAX = Y(I)
25 CONTINUE
IF (YMIN) 14,14,15
14 IGATE=IGATE+1
GO TO 38
15 CONTINUE

C
C
DETERMINE APPROPRIATE SCALING
CALL NORMAL(YMIN,IBASEXP)
YMINVAL = 10.0**IBASEXP
YMINLOG = LOG(YMINVAL)
ARG = YMAX
CALL NORMAL(ARG,MAXEXP)
IF (ARG.EQ.1.0) GO TO 26
MAXEXP = MAXEXP + 1
26 NSCALES = MAXEXP - IBASEXP
FIPS = YLENGTH/NSCALES
27 DY = FIPS/LOGF(10.0)

C
C
DRAW AND LABEL Y-AXIS
XOUT = -.08
XOUTT = -.20
XOUTN = -.15
TENLOCX = -.40
EXPLOCX = -.30
CALL PLOT (0,0,3)
DO 30 I=1,NSCALES
YLOC = YLOC + I*FIPS
IFXP = IBASEXP + 1
CALL PLOT (0,YLOC,2)
CALL PLOT (XOUT,YLOC,2)
CALL SYMBOL (TENLOCX,YLOC,TENSIZE,201,0,20)
EXPLOCY = YLOC + .7
CALL NUMBER (EXPLOCX,EXPLOCY,EXPsize,100,0,2012)
CALL PLOT (0,YLOC,3)
IF (1.EQ.NSCALES) GO TO 30
DO 29 J=2,9
YLOC = YLOC + LOGF(ELCATE(10))DY
CALL PLOT (0,YLOC,2)
CALL PLOT (XOUT,YLOC,2)
IF (FIPS.LT.2.0) GO TO 29
CALL NUMBER (XOUTN,YLOC,EXPsize,J,0,2011)

```

WHITNEY, MADER, AND ULRICH

```

29 CALL PLOT (0, YLOC, 3)
30 CONTINUE
   YLABLOC = YLENGTH*.5 - TENSIZ*NY*(3./7.)
   CALL SYMBOL(-.5,YLABLOC,TENSIZ,YLABEL,YC, NYC)
   GO TO 40

C   FOR NON-LOG SCALING OF Y
38 YLENGTH=YHEIGHT
   IF (YHEIGHT.GT.10.) YLENGTH=10.
C 39 CALL GETSCALE (Y,N*NY,YLENGTH,YMIN,DY,KY,YTICK,IFORM)
39 CALL GETSCALE (Y,N*NY,YLENGTH,YMIN,DY,KY,YTICK,IFORM)
   CALL AXIS (0,0,YLABEL,NYC,YLENGTH,9,0,YTICK,YMIN,DY,IFORM)

C
C   DRAW THE PLOT TO SCALE, LEAVING THE CONTENTS OF ARRAYS X AND Y UNCHANGED
C
40 CONTINUE
45 GO TO (50,60,70,80), IGATE

C   LOG-LOG SCALING
50 DO 52 NY=1,NYS
   I1=(NY-1)*KYS+1
   XIN=(LOGF(X(I1))-XMINLOG)*DX
   YIN=(LOGF(Y(I1))-YMINLOG)*DY
   CALL PLOT (XIN,YIN,3)
   DO 51 I=2,N
   IX=(I-1)*KX+1
   IY=(I-1)*KY+1
   XIN=(LOGF(X(IX))-XMINLOG)*DX
   YIN=(LOGF(Y(IY))-YMINLOG)*DY
51 CALL PLOT (XIN,YIN,2)
52 IF (NYS.GT.1) CALL SYMBOL (XIN,YIN,H,LINELABL(NY),G,8)
   RETURN

C   SEMILOGX SCALING
60 DY=YTICK/DY
   DO 62 NY=1,NYS
   I1=(NY-1)*KYS+1
   XIN=(LOGF(X(I1))-XMINLOG)*DX
   YIN=(Y(I1)-YMIN)*DY
   CALL PLOT (XIN,YIN,3)
   DO 61 I=2,N
   IX=(I-1)*KX+1
   IY=(I-1)*KY+1
   XIN=(LOGF(X(IX))-XMINLOG)*DX
   YIN=(Y(IY)-YMIN)*DY
61 CALL PLOT (XIN,YIN,2)
62 IF (NYS.GT.1) CALL SYMBOL (XIN,YIN,H,LINELABL(NY),G,8)
   RETURN

C   SEMILOGY SCALING
70 DX=XTICK/DX
   DO 72 NY=1,NYS
   I1=(NY-1)*KYS+1
   XIN=(X(I1)-XMIN)*DX
   YIN=(LOGF(Y(I1))-YMINLOG)*DY
   CALL PLOT (XIN,YIN,3)
   DO 71 I=2,N
   IX=(I-1)*KX+1
   IY=(I-1)*KY+1
   XIN=(X(IX)-XMIN)*DX
   YIN=(LOGF(Y(IY))-YMINLOG)*DY
71 CALL PLOT (XIN,YIN,2)
72 IF (NYS.GT.1) CALL SYMBOL (XIN,YIN,H,LINELABL(NY),G,8)
   RETURN

```

NRL REPORT 8074

```

C      LINEAR SCALING
      DO 82 NY=1,NYS
      DX=XTICK/DX
      DY=YTICK/DY
      I1=(NY-1)*NYS+1
      XIN=(X(1)-XMIN)*DX
      YIN=(Y(I1)-YMIN)*DY
      CALL PLOT (XIN,YIN,3)
      DO 81 I=2,N
      IX=(I-1)*KX+1
      IY=(I-1)*KY+1
      XIN=(X(IX)-XMIN)*DX
      YIN=(Y(IY)-YMIN)*DY
      81 CALL PLOT (XIN,YIN,2)
      82 IF (NYS.GT.1) CALL SYMBOL (XIN,YIN,0,LINE,LABL(NY),0,5)
      RETURN
      END

```

WHITNEY, MADER, AND ULRICH

```

SUBROUTINE FASTFOUR(A,M,INV,S,IFS,IFERR)
FASTFOUR ... DISCRETE FOURIER TRANSFORM ... FORTRAN 63
INPUT PARAMETERS TO BE SET BY USER BEFORE ENTERING FASTFOUR

A IS A 3-DIMENSIONAL ARRAY OF COMPLEX COEFFICIENTS, OF
DIMENSION (N(1),N(2),N(3)).
THE A'S ARE STORED WITH REAL PART OF A(I1,I2,I3) IN THE
LOCATION WITH INDEX 2*(I3*N1*N2 + I2*N1 + I1)+1 AND THE
IMAGINARY PART IN THE LOCATION IMMEDIATELY FOLLOWING.
IF THE FOURIER SERIES IS REQUESTED, ARRAY A IS REPLACED
BY ....
X(J1,J2,J3)=SUM A(K1,K2,K3)*W1**IK1*J1)*W2**IK2*J2)*W3**IK3*J3)
SUMMED OVER K1=0, N1-1; K2=0, N2-1; K3=0, N3-1.
WHERE W1 = N1 -TH ROOT OF UNITY.

N1=2**N(1) IS THE NO. OF POINTS IN THE 1TH DIMENSION.
THE DIMENSION OF A IN THE CALLING PROGRAM SHOULD BE TWICE
THE NUMBER OF COMPLEX ELEMENTS IN THE LARGEST A ARRAY TO
BE PROCESSED.
THE COMPLEX X'S ARE STORED IN THE SAME MANNER AS A.

IF THE FOURIER TRANSFORM IS REQUESTED, THE ARGUMENT A
IS TAKEN TO BE X AND IS REPLACED BY THE ARRAY A SATISFY-
ING THE FOURIER SERIES.

LET MT=MAX(M1,M2,M3)-2, NT=2**MT, WITH M BEING THE
M GIVEN WHEN THE TABLES ARE SET.
S(J)=SIN(J*PI/2 *NT) , J=1, NT-1

INV(J+1)=WORD CONTAINING BITS OF J IN INVERTED ORDER IN ITS
RIGHTMOST MT BIT POSITIONS, FOR J=0, NT-1

TO SET UP SIN AND INV TABLES ...
CALL SETUP(A,M,INV,S,0,IFERR)
ONE NEED NOT REPEAT THE CALL TO SETUP IF ONE DOES NOT
CHANGE THE MAXIMUM N.

DIMENSION A(1),N(1),INV(1),S(1),N(1),N(1)
EQUIVALENCE (N1,N(1)),(N2,N(2)),(N3,N(3))
CALL WITH IFS = +1 FOR FOURIER SERIES
CALL WITH IFS = -1 FOR FOURIER TRANSFORM.
CALL SETUP(A,M,INV,S,0,IFERR) TO SET UP TABLES OF SIN AND INV

IFERR = 0 WHEN ARGUMENTS OK AND OK
IFERR = 1 WHEN THERE IS AN ERROR IN CALLING

DATA (SIN45 = .7071067812), (PI=.31415926536)
IF(IFS) 12,13
MTT = XMAXOF(N(1),N(2),N(3)) -2
IF(MTT .LT. MT) 14,13
IFERR = 1 GO TO 9999
IFERR = 0
M1 = M(1) 1 M2 = M(2) 2 M3 = M(3)
N1 = 2**M1 3 N2 = 2**M2 4 N3 = 2**M3
IF(IFS) 16,9999,20
(CALCULATE TRANSFORM ... REPLACE A BY CONJUGATE)
NTOT= N1*N2*N3 5 FN = 1/ELCATEDUCLD 6 NTOT2=NTOT2
DO 18 I = 2,NTOT2,2
A(I-1) = A(I-1)*FN
A(I) = -A(I)*FN
NP(1)= N1*N1
NP(2) = NP(1)*N2
NP(3) = NP(2)*N3
HERE BEGINS THE LOOP OVER THE 3-DIMENSIONAL

```

NRL REPORT 8074

```

DO 250 ID = 1,3                                FASTF 20
IL = NP(3) - NP(ID)    $    IL1 = IL+1        FASTF 21
M1 = M(ID)                                FASTF 22
IF(1,NOT, M1) 250,30                          FASTF 23
30 IDIF = KBIT = NP(ID)                      FASTF 24
IF(M1.EQ. 2*(M1/2)) 60,40                     FASTF 25
C    M1 IS ODD. DO L = 1 CASE
40 KBIT = KBIT/2    $    KL = KBIT - 2        FASTF 26
DO 50 I = 1,IL1,IDIF                          FASTF 27
KLAST = KL + 1                                FASTF 28
DO 50 K = 1,KLAST,2                          FASTF 29
KD = K + KBIT                                FASTF 30
C    ONE STEP WITH L=1, J=0
C    A(K) = A(K) + A(KD)    $    A(KD) = A(K) - A(KD)
C    REAL PART
C    T = A(KD)    $    A(KD) = A(K) - T    $    A(K) = A(K) + T    FASTF 31
C    IMAG PART
C    T = A(KD+1)    $    A(KD+1) = A(K+1) - T    $    A(K+1) = A(K+1) + T    FASTF 32
50 CONTINUE                                FASTF 33
IF(M1.EQ. 1) 250,52                          FASTF 34
52 LFIRST = 3    $    JLAST = 1    $    GO TO 70    FASTF 35
C    DEF ... JLAST = 2*(L-2) - 1
60 LFIRST = 2    $    JLAST = 0    FASTF 36
70 DO 240 L = LFIRST, M1, 2                    FASTF 37
JJDIF = KBIT    $    KBIT = KBIT/4    $    KL = KBIT-2    FASTF 38
C    DO FOR J=0
DO 80 I = 1,IL1,IDIF                          FASTF 39
KLAST = 1+KL                                FASTF 40
DO 80 K = 1,KLAST,2                          FASTF 41
K1 = K+KBIT    $    K2 = K1+KBIT    $    K3 = K2+KBIT    FASTF 42
C
C    DO TWO STEPS WITH J=0
C    A(K) = A(K) + A(K2)    $    A(K2) = A(K) - A(K2)
C    A(K1) = A(K1) + A(K3)    $    A(K3) = A(K1) - A(K3)
C    A(K) = A(K) + A(K1)    $    A(K1) = A(K) - A(K1)
C    A(K2) = A(K2) + A(K3)*I    $    A(K3) = A(K2) - A(K3)*I
C
C    FIRST STEP REAL PART
C    T = A(K2)    $    A(K2) = A(K) - T    $    A(K) = A(K) + T    FASTF 43
C    T = A(K3)    $    A(K3) = A(K1) - T    $    A(K1) = A(K1) + T    FASTF 44
C    FIRST STEP IMAG PART
C    T = A(K2+1)    $    A(K2+1) = A(K+1) - T    $    A(K+1) = A(K+1) + T    FASTF 45
C    T = A(K3+1)    $    A(K3+1) = A(K1+1) - T    $    A(K1+1) = A(K1+1) + T    FASTF 46
C    SECOND STEP REAL PART
C    T = A(K1)    $    A(K1) = A(K) - T    $    A(K) = A(K) + T    FASTF 47
C    T = A(K3)    $    A(K3) = A(K2)+A(K3+1)    $    A(K2) = A(K2)-A(K3+1)    FASTF 48
C    SECOND STEP IMAG PART
C    T = A(K1+1)    $    A(K1+1) = A(K+1) - T    $    A(K+1) = A(K+1) + T    FASTF 49
C    T = A(K3+1)    $    A(K3+1) = A(K2+1) - T    $    A(K2+1) = A(K2+1) + T    FASTF 50
80 CONTINUE                                FASTF 51
IF(1,NOT, JLAST) 235,82                      FASTF 52
82 JJ = JJDIF + 1                            FASTF 53
C    DO FOR J=1
C    ILAST = IL+JJ
DO 85 I = JJ,ILAST,IDIF                      FASTF 54
KLAST = KL+1                                FASTF 55
DO 85 K = 1,KLAST,2                          FASTF 56
K1 = K + KBIT    $    K2 = K1 + KBIT    $    K3 = K2 + KBIT    FASTF 57
C
C    LETTING W = EXP(I*PI/4), W2 = W**2, W3 = W**3
C    A(K) = A(K) + A(K2)*I    $    A(K2) = A(K) - A(K2)*I
C    A(K1) = A(K1)*W + A(K3)*W3    $    A(K3) = A(K1)*W - A(K3)*W3
C
C    A(K) = A(K) + A(K1)    $    A(K1) = A(K) - A(K1)
C    A(K2) = A(K2) + A(K3)*I    $    A(K3) = A(K2) - A(K3)*I

```

WHITNEY, MADER, AND ULRICH

```

R = A(K2+1)      $      T = A(K2)                      FASTF 59
A(K2) = A(K) + R      $      A(K) = A(K) - R            FASTF 60
A(K2+1) = A(K+1) - T      $      A(K+1) = A(K+1) + T    FASTF 61
C
AWR = A(K1) - A(K1+1)      $      AWI = A(K1+1) + A(K1)  FASTF 62
R = A(K3) + A(K3+1)      $      T = A(K3) - A(K3+1)    FASTF 63
A(K3) = (AWR + R)*SIN45      $      A(K' ) = (AWR - R)*SIN45  FASTF 64
A(K3+1) = (AWI - T)*SIN45      $      A(K' ) = (AWI + T)*SIN45  FASTF 65
C
T = A(K1)      $      A(K1) = A(K1)-T      $      A(K) = A(K)+T    FASTF 66
T = A(K1+1)      $      A(K1+1) = A(K1+1)-T      $      A(K+1) = A(K+1)+T  FASTF 67
C
R = A(K3+1)      $      T = A(K3)                      FASTF 68
A(K3) = A(K2)+R      $      A(K2) = A(K2)-R            FASTF 69
A(K3+1) = A(K2+1)-T      $      A(K2+1) = A(K2+1)+T    FASTF 70
85  CONTINUE                                           FASTF 71
IF(JLAST,LE,1) 235,90                                FASTF 72
90  JJ=JJ+JJ01F                                         FASTF 73
C      NOW DO THE REMAINING J'S
DO 230 J = 2,JLAST                                     FASTF 74
C      FETCH W'S
C      DEF ... W1=EXP(1*PI/4)**INV(J)/NT , W2=W1**2, W3=W1**3
96  I = INV(J+1)                                           FASTF 75
98      W11=S(NT-I)      $      W12=S(I)                FASTF 76
      I2 = I+1      $      I2C = NT-I2                FASTF 77
      IF(I2C) 120,110,100                                FASTF 78
      I2 IS IN FIRST QUADRANT
100  W21 = S(I2C)      $      W22 = S(I2)      $      GO TO 130    FASTF 79
C      I2 IS PI/2
110  W21 = 0.      $      W22 = 1.      $      GO TO 130    FASTF 80
C      I2 IS IN SECOND QUADRANT
120  I2CC = I2C + NT      $      I2C = -I2C                FASTF 81
      W21 = -S(I2C)      $      W22 = S(I2CC)              FASTF 82
130  I3 = I+I2      $      I3C = NT-I3                    FASTF 83
      IF(I3C) 160,150,140                                FASTF 84
      I3 IN FIRST QUADRANT
140  W31 = S(I3C)      $      W32 = S(I3)      $      GO TO 200    FASTF 85
C      I3 = PI/2
150  W31 = 0.      $      W32 = 1.      $      GO TO 200    FASTF 86
C      I3 IN 2ND OR 3RD QUADRANT
160  I3CC = I3C + NT                                           FASTF 87
      IF(I3CC) 190,180,170                                FASTF 88
      I3 IN 2ND QUADRANT
170  I3C = -I3C                                           FASTF 89
      W31 = -S(I3C)      $      W32 = S(I3CC)      $      GO TO 200    FASTF 90
C      I3 = PI
180  W31 = -1.      $      W32 = 0.      $      GO TO 200    FASTF 91
C      I3 IN 3RD QUADRANT
190  I3CCC = NT+I3CC      $      I3CC = -I3CC                FASTF 92
      W31 = -S(I3CCC)      $      W32 = -S(I3CC)              FASTF 93
200  ILAST = IL+JJ                                           FASTF 94
      DO 220 I = JJ,ILAST,101F                             FASTF 95
      KLAST = KL+I                                           FASTF 96
      DO 220 K = I,KLAST,7                                   FASTF 97
      K1 = K+K01T      $      K2 = K1+K01T      $      K3 = K2+K01T    FASTF 98
C
C      DO TWO STEPS WITH J NOT 0
C
C      A(K) = A(K) + A(K2)*W2,      A(K2) = A(K) - A(K2)*W2
C      A(K1) = A(K1)*W1 + A(K3)*W3,      A(K3) = A(K1)*W1 - A(K3)*W3
C
C      A(K) = A(K) + A(K1)      $      A(K1) = A(K) - A(K1)
C      A(K2) = A(K2) + A(K3)*I      $      A(K3) = A(K2) - A(K3)*I
C
R = A(K2)*W21 - A(K2+1)*W22                                FASTF 99
T = A(K2)*W22 + A(K2+1)*W21                                FASTF 100

```

NRL REPORT 8074

```

A(K2) = A(K)-R      S   A(K) = A(K)+R      FASTF101
A(K2+1) = A(K+1)-T  S   A(K+1) = A(K+1)+T  FASTF102

C
R = A(K3)*W31 - A(K3+1)*W32      FASTF103
T = A(K3)*W32 + A(K3+1)*W31      FASTF104
AWR = A(K1)*W11 - A(K1+1)*W12    FASTF105
AWI = A(K1)*W12 + A(K1+1)*W11    FASTF106
A(K3) = AWR-R      S   A(K1) = AWR+R      FASTF107
A(K3+1) = AWI-T      S   A(K1+1) = AWI+T    FASTF108
T = A(K1)      S   A(K1) = A(K1)-T      S   A(K1) = A(K1)+T  FASTF109
T = A(K1+1)      S   A(K1+1) = A(K1+1)-T  S   A(K1+1) = A(K1+1)+T  FASTF110
R = -A(K3+1)      S   T = A(K3)          FASTF111
A(K3) = A(K2)-R      S   A(K2) = A(K2)+R    FASTF112
A(K3+1) = A(K2+1)-T  S   A(K2+1) = A(K2+1) +T  FASTF113
220 CONTINUE      FASTF114
C      END OF I AND K LOOPS
230 JJ = JJDIF+JJ      FASTF115
C      END OF J LOOP
235 JLAST = 4*JLAST +3      FASTF116
240 CONTINUE      FASTF117
C      END OF L LOOP
250 CONTINUE      FASTF118
C      END OF ID LOOP
C
C      WE NOW HAVE THE COMPLEX FOURIER SUMS BUT THEIR ADDRESSES ARE
C      BIT-REVERSED. THE FOLLOWING ROUTINE PUTS THEM IN ORDER....
NTSQ = NT*NT      FASTF119
IF(M3 .LT. MT) 370,360      FASTF120
360 N3VNT = N3/NT      S   MINN3 = NT      FASTF121
IG03 = 0      S   GO TO 380      FASTF122
370 N3VNT = 1      S   NTVN3 = NT/N3      S   MINN3 = N3      FASTF123
IG03 = 1      FASTF124
380 JJD3 = NTSQ/N3      FASTF125
IF(M2 .LT. MT) 470,460      FASTF126
460 N2VNT = N2/NT      S   MINN2 = NT      FASTF127
IG02 = 0      S   GO TO 480      FASTF128
470 N2VNT = 1      S   NTVN2 = NT/N2      S   MINN2 = N2      FASTF129
IG02 = 1      FASTF130
480 JJD2 = NTSQ/N2      FASTF131
IF(M1 .LT. MT) 570,560      FASTF132
560 N1VNT = N1/NT      S   MINN1 = NT      FASTF133
IG01 = 0      S   GO TO 580      FASTF134
570 N1VNT = 1      S   NTVN1 = NT/N1      S   MINN1 = N1      FASTF135
IG01 = 1      FASTF136
580 JJD1 = NTSQ/N1      FASTF137
600 JJ3 = J = 1      FASTF138
DO 880 JPP3 = 1,N3VNT      FASTF139
IPP3 = INV(JJ3)      FASTF140
DO 870 JP3 = 1,MINN3      FASTF141
IF(IG03) 620,610      FASTF142
610 IP3 = INV(JP3)*N3VNT      S   GO TO 630      FASTF143
620 IP3 = INV(JP3)/NTVN3      FASTF144
630 I3 = (IPP3+IP3)*N2      FASTF145
700 JJ2 = 1      FASTF146
DO 870 JPP2 = 1,N2VNT      FASTF147
IPP2 = INV(JJ2)+I3      FASTF148
DO 860 JP2 = 1,MINN2      FASTF149
IF(IG02) 720,710      FASTF150
710 IP2 = INV(JP2)*N2VNT      S   GO TO 730      FASTF151
720 IP2 = INV(JP2)/NTVN2      FASTF152
730 I2 = (IPP2+IP2)*N1      FASTF153
800 JJ1 = 1      FASTF154
DO 860 JPP1 = 1,N1VNT      FASTF155
IPP1 = INV(JJ1) + I2      FASTF156
DO 850 JP1 = 1,MINN1      FASTF157
IF(IG01) 820,810      FASTF158

```

WHITNEY, MADER, AND ULRICH

```

810 IP1 = INV(JP1)*N1VNT      *   GO TO 830
820 IP1 = INV(JP1)/NTVN1
830 I = 2*(IP1+IP1)+1
    IF(J.GE. 1) 845,840
840 T = A(I)      *   A(I) = A(J)      *   A(J) = T
    T = A(I+1)    *   A(I+1) = A(J+1)  *   A(J+1) = T
845 J = J+2
850 CONTINUE
860 JJ1 = JJ1 + JJD1
    C      END OF JPP1 AND JP2 LOOPS
870 JJ2 = JJ2 + JJD2
    C      END OF JPP2 AND JP3 LOOPS
880 JJ3 = JJ3 + JJD3
    C      END OF JPP3 LOOP
    C      WAS THIS A TRANSFORM....
    IF(-IFS) 9999,9999,887
    YES, REPLACE A BY CONJUG(A).
882 DO 884 I = 2,NTOT2+2
884 A(I) = -A(I)
    GO TO 9999
    C      THAT'S THE END .....
    ENTRY SETUP
    C      THIS PROGRAM COMPUTES THE SIN AND INV TABLES
900 MT = XMAXOF(2,(XMAXOF(M(1),M(2),M(3))-2))
904 IF(MT.LE. 1) 906,905
905 IFERR = 1      *   GO TO 9999
906 IFERR = 0
    NT = 2*MT      *   NTV2 = NT/2
    C      SET UP SIN TABLE
    C      THETA = PI/2*(L+1) ... FOR L=1 ...
    THETA = .25*PI
    C      JSTEP = 2*(MT-L+1) ... FOR L=1 ...
    JSTEP = NT
    C      JDIF = 2*(MT-L) ... FOR L=1 ...
    JDIF = NTV2
    C
    S(JDIF) = SIN(THETA)
    DO 950 L = 2,MT
    THETA = .5*THETA
    JSTEP2 = JSTEP      *   JSTEP = JDIF      *   JDIF = JSTEP/2
    S(JDIF) = SIN(THETA)
    JCI = NT-JDIF
    S(JCI) = COS(THETA)
    JLAST = NT-JSTEP2
    IF(JLAST.LT. JSTEP) 950,920
920 DO 940 J = JSTEP,JLAST,JSTEP
    JC=NT-J      *   JD=J+JDIF
940 S(JD) = S(J)*S(JCI) + S(JDIF)*S(JC)
950 CONTINUE
    C
    C      SET UP INV(J) TABLE
    C
    C      MLEXP = 2*(MT-L) ... FOR L=1 ...
960 MLEXP = NTV2
    C      LMIEXP = 2*(L-1) ... FOR L=1 ...
    LMIEXP = 1
    INV(1) = 0
    DO 980 L=1,MT
    INV(LMIEXP+1) = MLEXP
    DO 970 J = 2,L+1,1EXP
    JJ=J+LMIEXP

```

FASTF159
FASTF160
FASTF161
FASTF162
FASTF163
FASTF164
FASTF165
FASTF166
FASTF167

FASTF168
FASTF169

FASTF170
FASTF171
FASTF172
FASTF173

FASTF174

FASTF175
FASTF176
FASTF177
FASTF178
FASTF179

FASTF180
FASTF181
FASTF182

FASTF183
FASTF184
FASTF185
FASTF186
FASTF187
FASTF188
FASTF189
FASTF190
FASTF191
FASTF192
FASTF193
FASTF194
FASTF195

FASTF196

FASTF197
FASTF198
FASTF199
FASTF200
FASTF201
FASTF202

NRL REPORT 8074

```

970  INV(J) = INV(J)+%TLEXP
      %TLEXP = %TLEXP/2      L1LEXP = L1LEXP/2
980  CONTINUE
999  RETURN

```

END

FASTF203
FASTF204
FASTF205
FASTF206

FAST 207

[illegible]