

AD-A103 285

UNITED TECHNOLOGIES CORP WEST PALM BEACH FLA
SYSTEM OPTICAL QUALITY USERS GUIDE. PART 2.(U)
MAR 80 J L FORGHAM, S S TOWNSEND

F/6 20/5

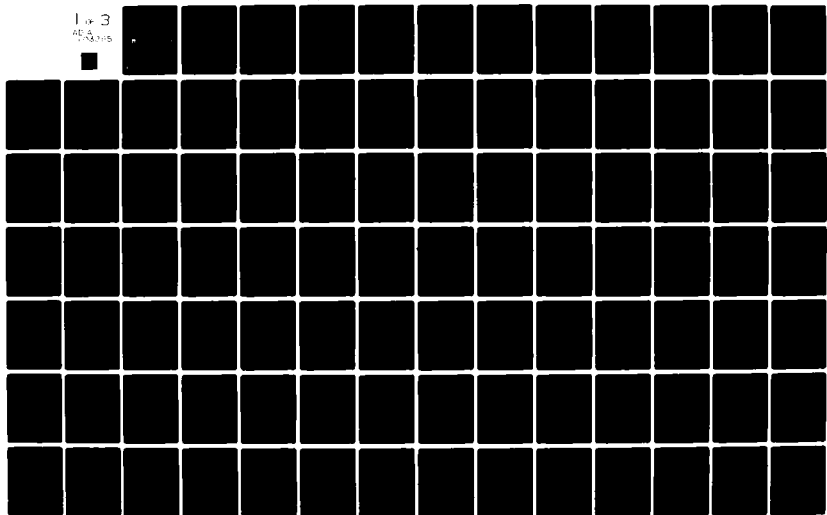
F29601-77-C-0025

UNCLASSIFIED

AFNL-TR-79-141-PT-2

NL

1 of 3
454
03/08/85



② LEVEL IV

AD-E200 723

SYSTEM OPTICAL QUALITY USERS GUIDE

Part 2

J.L. Forgham S. S. / Townsend J. L. / Campbell

United Technologies Corporation
West Palm Beach, FL 33402

Mar 1980

DTIC ELECTE
AUG 24 1981

Final Report

B

Approved for public release; distribution unlimited.

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

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

389121

81 5 20 055

AD A103285



DTIC FILE COPY

This final report was prepared by the United Technologies Corporation, West Palm Beach, Florida, under Contract F29601-77-C-0025, Job Order 00011408 with the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico. Captain J. Dale Holt (ARLO) was the Laboratory Project Officer.

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report have been authored by a contractor of the United States Government. The United States Government retains a nonexclusive, royalty-free license to publish or reproduce the material contained herein, or allow others to do so, for the United States Government purposes.

This report has been reviewed by the Public Affairs Office and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

J. Dale Holt
J. DALE HOLT
Captain, USAF
Project Officer

James L. Stapp
JAMES L. STAPP
LtCol, USAF
Chief, Optical Systems Branch

FOR THE DIRECTOR
John A. Carpenter
JOHN A. CARPENTER
Colonel, USAF
Deputy Chief, Laser Development Division

DISCLAIMER NOTICE

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

SOQ USER GUIDE UPDATES

June 1980 Updates to SOQ80128

INTRODUCTION

This document defines the changes made to the SOQ code (SOQ80128) between January and June of 1980. The changes either correct shortcomings found in the code or, more usually, document the increased capability being continually built into the code. The SOQ code is maintained as SOQ80128 June PL, ID = AFLOJRA as a NOS/BE-1 CDC update format file.

UPDATES

1. *ID FIXZRN

This update redefines the coefficients to be input to the Zernike subroutine. This new convention is more physically meaningful in that, at least for lower orders, the coefficients are in waves. For example, to impose one wave peak to peak of defocus (P_4) on a beam, one would input $P(4)=1$. The phase applied is now:

$$\phi(I,J) = \sum_k P_k \pi Z_k(I,J)$$

The subroutine affected is ZERN. This update does not effect the rest of the code.

2. *ID FIXJTR

This update ensures a correct definition of DF in subroutine JITRBC since when JITRBC is called from subroutine QUAL, the X-coordinate array contains $R\lambda/D$ coordinates, not the spatial coordinates.

Only one line of the code is affected by this update.

3. *ID ROTZRN

Due to different coordinate system orientations for data, it became necessary to allow for this variation within subroutine ZERN.

Define the data x and y coordinates to be XROT and YROT, and the SOQ x and y coordinates to be XIN and YIN. The rotation angle is then defined to be θ (in radians).

June 1980 Updates to SOQ80128

Page 2

$\text{COSROT} = \text{COS}(\theta)$

$\text{SINROT} = \text{SIN}(\theta)$

$\text{XROT} = \text{XIN} \times \text{COSROT} + \text{YIN} \times \text{SINROT}$

$\text{YROT} = -\text{XIN} \times \text{SINROT} + \text{YIN} \times \text{COSROT}$

Application of Zernike polynomials to and SOQ point located at (XIN, YIN) would then be calculated using Z(XROT, YROT). The possibility of axis flips are also accounted for and are flagged by FLIPX or FLIPY not equal to zero. Namelist ZERNS is modified to include FLIPX, FLIPY and the rotation angle (in degrees) ZTHETA. No common was modified. This update modified only subroutines GDL and ZERN.

*IDENT FIXZRN

*/ ZERN

```
*DELETE ZRNKE.11E
      DEL = CFL*3.14159264
*DELETE ZRNKE.12E
      C 2(X,2) FFI(N) = FI*F(N)*2(N) //
```

*IDENT FIXLTR

*/ JITREG

```
*DELETE JITTER.29, JITTER.30
      CF = 1./(FLCAT(NPTS)*CY)
```

*IDENT ROTZRN

*/ GCL

```
*DELETE ZRNINFC.3
      NAMELIST /ZERN/ PC, F, FFPAG, SIGMAY, XTERMZ, ZTHETA, FLIPX, FLIFY
*INSERT ZRNKE.5
C      ZTHETA = THE CLOCKWISE ANGLE OF ROTATION OF THE DECOMPOSITE
C      AXES ONTO THE SGG COORDINATE SYSTEM
C      BEFORE CALCULATION OF THE ZERNKE POLYNOMIALS.
C      IT IS INPUT IN DEGREES.
C      FLIPX = 1. RESULTS IN A FLIP ABOUT THE X AXIS BEFORE
C      ROTATION.
C      FLIFY = 1. RESULTS IN A FLIP ABOUT THE Y AXIS BEFORE
C      ROTATION.
*DELETE ZRNINFC.2
      DIMENSION FZ2SV(20,10)
*INSERT ZRNINFC.7
      ZTHETA = 0.
      FLIPX = 0.
      FLIFY = 0.
*INSERT ZRNINFC.9
      FZ2SV(IZERN,3) = ZTHETA*3.141593/180.
      FZ2SV(IZERN,4) = FLIPX
      FZ2SV(IZERN,5) = FLIFY
*DELETE ZRNINFC.10, ZRNINFC.11
      244 CALL ZERN(FZ2SV(IZERN,1), FZ2SV(IZERN,2), FZ2SV(IZERN,3),
X          FZ2SV(IZERN,4), FZ2SV(IZERN,5),
X          FZSAVE(25, IZERN), FZSAVE(1, IZERN))
```

*/ ZERN

```
*DELETE ZRNINFC.12
      SLEROLTIME ZERN(SIGMAY, XTERMZ, THETA, FLIPX, FLIFY, PC, F)
*INSERT ZRNKE.72
      CCSRCT = COS(THETA)
      SINRCT = SIN(THETA)
*DELETE ZRNKE.75
*DELETE ZRNKE.77
      XIA = X(IX)
      YIA = X(IY)
      IF(FLIPY.GT..5) YIA=-YIA
      IF(FLIPX.GT..5) XIA=-XIA
      YRCT = XIA*CCSRCT + YIA*SINRCT
      YRCT = -XIA*SINRCT + YIA*CCSRCT
      IF(FLIPX.LT.-.5) YRCT=-YRCT
      IF(FLIPY.LT.-.5) XRCT=-YRCT
      XSG = XRCT**2
      YSG = YRCT**2
*DELETE ZRNKE.80
      THET = ATAN2(YRCT, YRCT)
```

*IDENT MORSUM

*INSERT SUMMARY.515

C

C **** COPY TAPE(50) TO CLTFLT:

C

END FILE 50

C

WRITE(6,3035)

REWIND 50

7000 READ(50,4005) IC1,C2

4005 FORMAT(11,21A4)

IF(EOF(50).NE.0.) GO TO 7015

C IF(IC1.EG.1) WRITE(6,3035)

WRITE(6,4040) C2

4040 FORMAT(10X,21A4)

GO TO 7000

7015 REWIND 50

WRITE(6,3035)

C

REWIND 57

4000 READ(57,4005) IC1,C2

IF(EOF(57).NE.0.) GO TO 4015

IF(IC1.EG.1) WRITE(6,3035)

WRITE(6,4040) C2

GO TO 4000

4015 REWIND 57

WRITE(6,3035)

C

REWIND 57

6000 READ(57,4005) IC1,C2

IF(EOF(57).NE.0.) GO TO 6015

IF(IC1.EG.1) WRITE(6,3035)

WRITE(6,4040) C2

GO TO 6000

6015 REWIND 57

WRITE(6,3035)

C

C **** COPY TAPE(ISUMRY) TO CLTFLT:

C

REWIND ISUMRY

5000 READ(ISUMRY,3005) IC1,C2,C3

IF(EOF(ISUMRY).NE.0.) GO TO 5015

IF(IC1.EG.1) WRITE(6,3035)

WRITE(6,3040) C2,C3

GO TO 5000

5015 REWIND ISUMRY

WRITE(6,3035)

C

C **** COPY TAPE(50) TO CLTFLT:

C

WRITE(6,3035)

REWIND 50

8000 READ(50,4005) IC1,C2

IF(EOF(50).NE.0.) GO TO 8015

C IF(IC1.EG.1) WRITE(6,3035)

WRITE(6,4040) C2
CC TO 9000
9018 REWIND 50
WRITE(6,3038)
C

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWL-TR-79-141	2. GOVT ACCESSION NO. <i>DA-A103 285</i>	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SYSTEM OPTICAL QUALITY USERS GUIDE Part 2 of 3.		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. L. Forgham S. S. Townsend J. L. Campbell		8. CONTRACT OR GRANT NUMBER(s) F29601-77-C-0025
9. PERFORMING ORGANIZATION NAME AND ADDRESS United Technologies Corporation West Palm Beach, FL 33402		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 00011408/63605F
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (ARLO) Kirtland Air Force Base, NM 87117		12. REPORT DATE March 1980
		13. NUMBER OF PAGES 262
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION, DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This report is divided into three parts. Part 1 consists of the front matter and text pages 1-34. Part 2 consists of text pages 35-296 and the References. Part 3 consists of Appendices A and B and distribution list pages 297-360.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Laser Optical System Code High Power Laser Optics High Energy Laser Optical Quality		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the System Optical Quality (SOQ) code structure and the input to the code required for analyzing High Power Laser Optical Systems. The SOQ code provides the designer with a physical optics model of the system. The code traces the beam from its point of origin in the resonator through the optical train into the far field. This report is divided into three parts. Part 1 describes the general structure of the SOQ code and establishes a correlation between the usual optical elements encountered in the optical		

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Continued).

train/gas dynamic laser resonator and the appropriate SOQ models. Part 2 acquaints the user with the individual SOQ subroutines and their analytical formulations as manifested in Fortran within the SOQ framework. It also delineates the input required to exercise the subroutines, familiarizes the user with the operation of the SOQ model, and contains working input modules which carry the user through the usual calculations of the SOQ code from input generation to loaded cavity calculations. Part 3 contains Appendices describing SOQ updates.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	23 cp

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECTION III
MAIN EXECUTIVE CODE

1. PROGRAM SOQ

a. Purpose -- Program SOQ is the main driver program or executive code for the total SOQ code. Many parameters such as mesh size, number of points, initial position of the optical axis, the initial coordinate array, and the initial field itself are established in this routine. Once the above parameters have been initialized, there are several options available for operations on the field. Those available are:

- (1) Calling subroutine GDL, the executive program for propagating the field through the optical elements
- (2) Performing a quality calculation
- (3) Gradient search optimization
- (4) Parametric studies.

The above options can be activated in any order and as many times as desired by successive reads of namelist START. The flag for ending execution of the entire deck is to set WWL = 0 in the last read of START.

b. Formalism -- The only major explicit calculations done in SOQ are those which determine the initial field when it is not to be read in. The OPTIONS are:

- (1) Plane wave - constant amplitude
- (2) Plane wave - Gaussian amplitude
- (3) Spherical wave - constant amplitude
- (4) Spherical wave - Gaussian amplitude.

Letting $E(x,y)$ represent the field, $A(x,y)$ the field amplitude, and $\phi(x,y)$ the field phase, then the field is determined by:

$$E(x,y) = A(x,y) e^{i\phi(x,y)} \quad (8)$$

where

$$A(x,y) = \frac{\text{const.}}{\text{const.}} e^{-\left(\frac{x^2 + y^2}{\sigma^2}\right)} \quad (9)$$

and

$$\phi(x,y) = \begin{cases} 0 \\ e^{-\frac{\pi}{\lambda R} (x^2 + y^2)} \end{cases} \quad (10)$$

The other calculations based on input distributions are performed in subroutines.

c. Fortran -- The only common variables that are not altered in this routine are SPACE and CFIL. The others are altered and are defined as follows:

CU = the complex field array
 X = the coordinate array
 DRX = the x position of the optical axis
 DRY = the y position of the optical axis
 NPTS = the number of points in the x direction
 NPY = the number of points in the y direction
 = NPTS if SYMTRC is false
 = NPTS/2 if SYMTRC is true
 WL = the wavelength of the radiation
 PLTSG = plotting parameter (none, amplitude, or intensity)
 INT = set to 0

The relevant parameters are read into the program by means of the namelists described below.

(1) Namelist START -- This namelist is used to initialize parameters such as field, mesh, and coordinates, and is used to direct the calculation to other sections of SOQ. It is read repeatedly until WWL \leq 0 is encountered.

```

C.....C
C   NAMELIST /START/ #L, #CAL, #NPTS, #DUMX, #DUMY, #RSTRT, #N, #R, #CALL,
C   X, #MPGS, #ITNUM, #SYMC, #UGAUSS, #TITLES, #PHRAD
C   #, #PLOTS
C #PLOTS=#1.. AMPLITUDE, PHASE SLICE PLOTS
C #PLOTS=#0.. NO SLICE OR ISO-INTENSITY PLOTS
C #PLOTS=#1.. INTENSITY, PHASE SLICE PLOTS
C #PLOTS=#1.. INTENSITY, PHASE SLICE PLOTS          OT(IPLTS)
C   #CALL CONTROLS THE MOVEMENT INSIDE MAIN
C   #  = 2. GDL SECTION, CALLS GDL AND READ CU,X FROM DISK
C   #  = 3 CALL TO QUALITY ALGORITHM, READS QLOT
C   #  = 4 CALL TO ANY OF THE GOULD PLOTTING PACKAGES, READS THRED
C   #  = 5 STARTS OPTIMIZATION ALA JAVION, READS OPTIM
C   #  = 6 PARAMETRIC STUDIES.. INVOLVES CHANGING ARC ARRAY FOR GDL,
C   #  #READS PARAM
C.....C
C   #L IS RADIATION WAVELENGTH
C   #CAL IS INITIAL SIZE OF CALCULATION REGION
C   #NPTS IS NUMBER OF FIELD POINTS ACROSS #CAL
C   #DUMX, #DUMY = THE (X,Y) POSITION OF THE CENTER OF CU RELATIVE
C   #              TO THE OPTICAL AXIS
C   #RSTRT IS CONTROL FOR RESTARTING WITH EXISTING GAIN CO-EFF AND
C   #INITIAL FIELD FROM PREVIOUS RUN.....
C   #TRUE, IF RESTARTING, OR IF FIELD IS TO BE READ FROM #H
C   #FALSE, IF NOT, OR IF INITIAL FIELD AND XES ARE TO BE CALC
C   #N = UNIT NUMBER OF DATA SET FOR GDL AND CAVITY
C   #IF #N = 0, THEN THERE IS NO CALL TO GDL
C
C   #H IS UNIT NUMBER OF INPUT FIELD TO GDL
C   #IF #H = 0, THEN NOTHING IS READ
C   #AMPGS IS INITIAL AMPLITUDE OF STARTING BEAM: PEAK AMPLITUDE
C   #FOR GAUSSIAN
C   #PHRAD IS PHASE FRONT RADIUS OF CURVATURE (#0.0 FOR PLANE)
C   #ITNUM IS THE ITERATION NUMBER...IF UNSPECIFIED IT READS OFF DISK
C
C   #SYMC IS LOGICAL FOR SYMMETRIC ANALYSIS OR NOT
C   #UGAUSS IS DIAMETER AT WHICH GAUSSIAN AMPLITUDE = 1.0/F
C.....C

```

(2) Namelist QLOT -- This namelist establishes the parameters necessary to perform quality calculations.

```

C   NAMELIST /QLOT/ TITLE, #INT, #H, #ISAV, #PHASE, #H# ,#F
C
C   TITLE FOR PLOTS IN QUALITY ROUTINE
C   #INT IS PLOTTING PARAMETER FOR PLOT...QUALITY PLOTS
C   #  = 0 ISO-INTENSITY AND POWER VS #L/D GOULD PLOTS
C   #  = 1 FAR-FIELD PWR VS #L/D GOULD.....
C   #  = 2 NO GOULD PLOTS, CALCULATES POWER DIST. ONLY
C   #  = 3 ISO-INTENSITY GOULD, PWR DIST, BUT NO PWR #L/D PLOTS
C   #  = 4 CALC POWER INSIDE #H# ONLY...NO CALL TO PLOT
C   #H# = BEAM DIAMETER
C   #ISAV IS SAVING PARAMETER.....
C   #  = 0, DON'T SAVE          #  = 1, SAVE INPUT FIELD
C   #  = -1 USE DATA SET #9 FOR INPUT

```

```

C      IPHASE CONTROLS THE PHASE CORRECTIONS APPLIED TO THE FIELD
C      = 0 NONE
C      = 1 PLANAR CORRECTION
C      = 2 SPHERICAL
C      = 3 BOTH
C      RHR IS THE HUCKET SIZE FOR OPTIMAZATION...IF A CALL TO QUAL IS
C      DONE BEFORE OPTIMUM THE HUCKET IS SPECIFIED HERE
C      NR IS PL/DIAGNOSIS FOR QUALITY CALCULATION
C
C*****

```

(3) Namelist THRED -- THRED establishes the parameters required for three-dimensional plotting routines.

```

NAMLIST / THRED / PLOT3D, TITLE3, DIAM,
*      PLTISO, RPLOT, DIATSO, PSLICE, NP, JFAZE, AMAG
C
C      PLOT3D = .TRUE. FOR THREE DIMENSIONAL PLOTS OF BEAM FIELD
C      = .FALSE. FOR NO PLOTS
C      TITLE3 = TITLE INFORMATION FOR THREE DIMENSIONAL PLOTS
C      DIAM = DIAM OF ILLUSTRATED FIELD ON PLOTS
C
C      PLTISO IS LOGICAL FOR ISOPLOTS OF FIELD
C      RPLOT IS THE RADIUS OF CIRCLE DRAWN ON ISOPLOT FOR REFERENCE
C      DIATSO IS DIAMETER OF ISOPLOTS DESIRED
C      PSLICE IS LOGICAL FOR SLICE PLOTS OF FIELD
C      NP = THE SLICE IN Y-DIR. PLOTTED. IF = 0... NP = NPTS/2
C      JFAZE = 0. NO PHASE PLOT FOR THIS
C      = 1. GET THE PHASE
C
C*****

```

(4) Namelist OPTIM - Namelist OPT2 -- These two namelists are used by the optimization portion of the SOQ routine. OPTIM must be read first to direct the optimization procedure. OPT2 establishes which parameters are to be used in the procedure and their constraints.

```

NAMLIST/ OPTIM / RH, IPOT, NIND, NRIGHT, ORR
C      RH = HUCKET SIZE FOR QUALITY OPTIMIZATION
C      IPOT = 1 POWER WITHIN RH
C      = 2 TOTAL POWER IN BEAM
C      = 3 PEAK INTENSITY
C      NIND IS NUMBER OF IND VARIABLES TO BE OPTIMIZED
C      NRIGHT = BIGGEST NUMBER OF ITERATIONS TO BE PERFORMED
C      ORR IS THE BEAM DIAMETER FOR QUALITY CALC...IF CALLED TO QUAL
C      EARLIER THIS IS NOT NEEDED
C
C

```

```

NAMELIST/OPT2/ TEL1, TEL2, TEL3, XMIN, XMAX, XADD
C (TEL1,IFL2,TEL3) IS THE VECTOR DESCRIBING THE POSITION OF THE
C OPTIMIZED PARAMETER...IN OPERATIONAL SPACE
C XMIN AND XMAX ARE THE CONSTRAINTS ON THE OPTIMIZED VECTOR
C XADD IS A CONSTANT ADDED TO THE OPTIMIZED VARIABLE SUCH THAT
C ITS VALUE IS NEVER EQUAL TO ZERO
C THERE ARE NIND NUMBER OF CALLS TO THIS NAMELIST
C

```

(5) Namelist PARAM -- This namelist gives the parameters to be varied and what values are to be used.

```

NAMELIST / PARAM / NEL1,NEL2,NEL3, NPARA, XNPARA,
X MEL1,MEL2,MEL3, MPARA, XMPARA
C (NEL1,NEL2,NEL3) IS THE VECTOR DESCRIBING THE POSITION OF THE
C VARIABLE WHICH IS TO BE VARIED
C NPARA,MPARA ARE THE NUMBER OF CHANGES IN EACH VARIABLE
C XNPARA,XMPARA ARE THE ARRAYS THAT CONTAIN THE VALUES WHICH
C ARE TO BE USED
C *****IF ONLY ONE SET IS TO BE VARIED USE ONLY THE NPARA SET,
C AND SET MEL1 = 0. THE NCS ARE THE INNER LOOP *****
C IF AN ARRAY IS TO BE CHANGED AND NO CALL AT THIS TIME TO
C AUTO(GOLD), THEN SET NPARA=0...THEN TWO VALUES CAN BE CHANGED
C IF ONLY ONE IS TO BE CHANGED SET MEL1=0
C *****ALL CALLS BETWEEN GOLD AND PARAM TO QUAL.PLOT...WILL BE REPEATED
C INSIDE THE PARAMETRIC LOOP
C *****

```

(6) Program SOQ (Program SOQ Flow Chart (Fig. 12) appears on page 40.)

PROGRAM SOQ 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

PROGRAM SOQ(OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,	CUNR1	1
A TAPE6=OUTPUT,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12,TAPE13,	MAIN	3
B TAPE14,TAPE15,TAPE16,TAPE17,TAPE18,TAPE19,TAPE20,TAPE21,	MAIN	4
CTAPE22,TAPE23,TAPE24,TAPE25,TAPE26,TAPE27,TAPE28,TAPE29,	SOU77CY1	1
DTAPE30,TAPE31)	SOU77CY1	2
LEVEL 2+CU,CUN,SPACE	CUNR2	1
COMMON /PST/ SPACE(1000)	CUNR2	2
COMMON/MEL1/CU(16384),CFIL(16512),X(126),NL,NPTS,NPY,UNX,UNY	MAIN	7
COMMON /PLTISG/ PLTISG	LHOP1	1
COMMON /INITL/ INT	MAIN	8
DIMENSION TITLE(20),AS(3),XUP(4),XLOW(4),XUP(4),	MAIN	9
XLOP(3,4),XSCH(4),ABC(12,20,9),TITLE3(20),XUPADD(4),	CUOASTG	1
Z XMPARA(10), XNPARA(10), MAINC(25), TITLES(20), CUN(32/68)	MAIN	11
COMPLEX CU,CFIL,CUNS	MAIN	12
LOGICAL MESTMT,PLUT3D,PLTISU,PSLICE,CALQL,SYMFNC	MAIN	13
EQUIVALENCE (CU(1),CUN(1))	MAIN	14
DATA WL,OCAL,NNPTS,DUHA,DUHY/-1.0,0.0,0.0,0.0/	CUNR2	3
DATA DCAL,NESTR,IN,IB,NCALL,AMPGES,ITNUM,SYMFNC,UGAUSS,PHINAD	MAIN	15
X / 0.0, .THUE, 0, 8, 2, 1.0, -1, .FALSE, 0, 0, 0, 0 /	MAIN	16
DATA TITLES/20*H /	MAIN	17
DATA IQLT,UB,ISAV,IPHASE,NBH,NF /0.0,0.0,0.0,1.0,0.0,0/	MAIN	18
DATA TITLE/20*H /	MAIN	19
DATA MH,IMPI,NIND,NBIGIT,UMB /2.0,1.0,1.0,0/	MAIN	20
DATA PLUT3D,DIAM,PLTISU,NPLUT,UIAISU,PSLICE,NP,JFAZE,XMAU	MAIN	21
X /.FALSE,0.0, .FALSE,0.0, 0.0, .FALSE,0, 0,1.0/	MAIN	22
DATA PLUIS / 0. /	LHOP1	2
DATA TITLE3/20*H /	MAIN	23

EXECUTIVE ROUTINE STRUCTURE

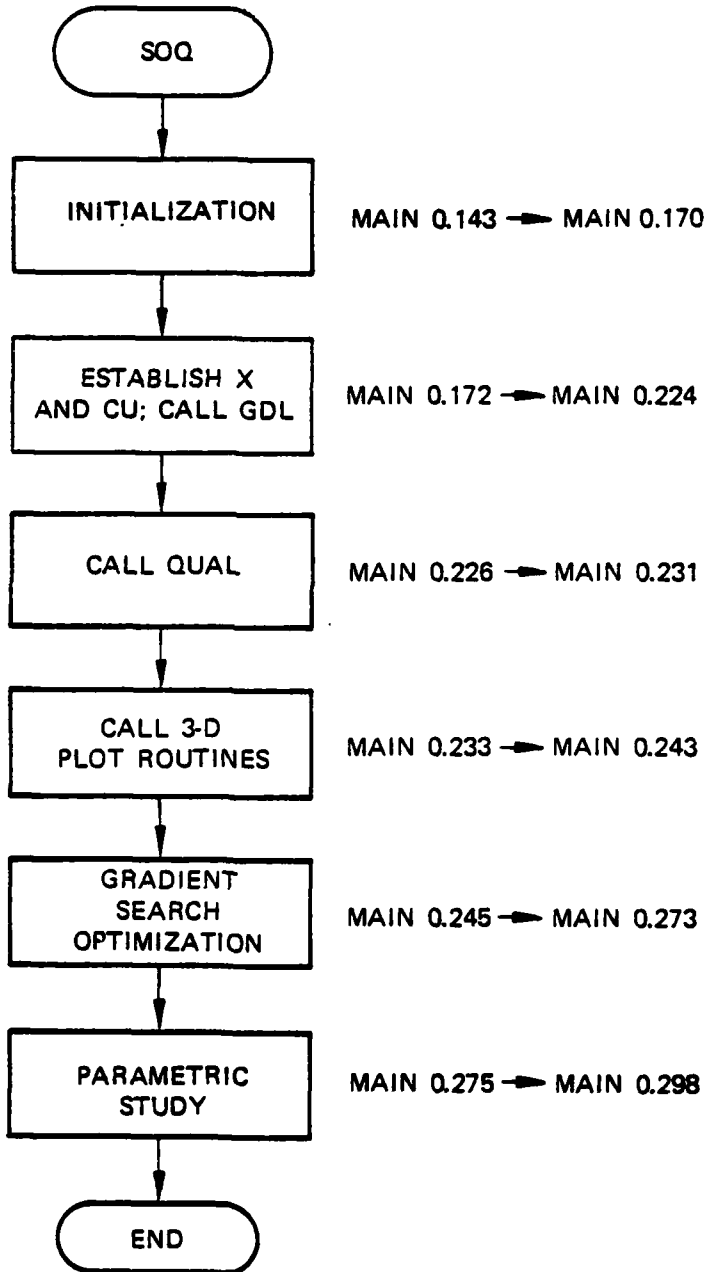


Figure 12. Program SOQ flow chart.

```

C.....C MAIN 24
  NAMELIST /STANT/ WVL,DCAL,NMPTS,DXA,DYR,HESTRT,IN,IB,NCALL,
  X AMPGES, ITNUM, SYMTRC ,DGAUSS , TITLE, PHIRAU
  X , PLOTS
C PLOTS=1., AMPLITUDE , PHASE SLICE PLOTS LHOPI 3
C PLOTS= 0., NO SLICE OR ISO-INTENSITY PLOTS LHOPI 4
C PLOTS= 1., INTENSITY, PHASE SLICE PLOTS LHOPI 5
C PLOTS=1., INTENSITY, PHASE SLICE PLOTS OT(IPLTS) LHOPI 6
C NCALL CONTROLS THE MOVEMENT INSIDE MAIN MAIN 7
C = 2, GUL SECTION,CALLS GUL AND HEAD CURV FROM DISK MAIN 27
C = 3 CALL TO QUALITY ALGORITHM, HEADS QLOT MAIN 28
C = 4 CALL TO ANY OF THE GOULD PLOTTING PACKAGES, HEADS TMED MAIN 29
C = 5 STARTS OPTIMAZATION ALA DAVIDON, HEADS OPTIM MAIN 30
C = 6 PARAMETRIC STUDIES..INVOLVES CHANGING ABC ARRAY FOR GUL, MAIN 31
  HEADS PARAM MAIN 32
C.....C MAIN 33
C WVL IS RADIATION WAVELENGTH MAIN 34
C DCAL IS INITIAL SIZE OF CALCULATION REGION MAIN 35
C NMPTS IS NUMBER OF FIELD POINTS ACROSS UCAL MAIN 36
C DXA,DYR = THE (X,Y) POSITION OF THE CENTER OF CU RELATIVE MAIN 37
  TO THE OPTICAL AXIS MAIN 38
C HESTRT IS CONTROL FOR RESTARTING WITH EXISTING GAIN CO-EFF AND MAIN 39
  INITIAL FIELD FROM PREVIOUS RUN..... MAIN 40
C .TRUE. IF RESTARTING, OR IF FIELD IS TO BE HEAD FROM IB MAIN 41
C .FALSE. IF NOT, OR IF INITIAL FIELD AND ACS ARE TO BE CALC MAIN 42
C IN = UNIT NUMBER OF DATA SET FOR GUL AND CAVITY MAIN 43
C IF IN = 0, THEN THERE IS NO CALL TO GUL MAIN 44
C MAIN 45
C IB IS UNIT NUMBER OF INPUT FIELD TO GUL MAIN 46
C IF IB = 0, THEN NOTHING IS HEAD MAIN 47
C MAIN 48
C AMPGES IS INITIAL AMPLITUDE OF STARTING BEAM (PEAK AMPLITUDE MAIN 49
  FOR GAUSSIAN) MAIN 50
C PHIRAU IS PHASE FRONT RADIUS OF CURVATURE (=0.0 FOR PLANE) MAIN 51
C ITNUM IS THE ITERATION NUMBER...IF UNSPECIFIED IT HEADS OFF DISK MAIN 52
C MAIN 53
C SYMTRC IS LOGICAL FOR SYMMETRIC ANALYSIS OR NOT MAIN 54
C DGAUSS IS DIAMETER AT WHICH GAUSSIAN AMPLITUDE = 1.0/E MAIN 55
C MAIN 56
C.....C MAIN 57
C MAIN 58
C MAIN 59
C NAMELIST/QLOT/ TITLE, IQLT, UB, ISAV, IPHASE, HBB ,HF MAIN 60
C TITLE FOR PLOTS IN QUALITY ROUTINE MAIN 61
C IQLT IS PLOTTING PARAMETER FOR PLOT...QUALITY PLOTS MAIN 62
C = 0 ISO-INTENSITY AND POWER VS HL/D GOULD PLOTS MAIN 63
C = 1 PAN-FIELD PWR VS HL/D GOULD..... MAIN 64
C = 2 NO GOULD PLOTS CALCULATES POWER DIST. ONLY MAIN 65
C = 3 ISO-INTENSITY GOULD, PWR DIST, BUT NO PWR HL/D PLOTS MAIN 66
C = 4 CALC POWER INSIDE HBB ONLY...NO CALL TO PLOT MAIN 67
C UB = BEAM DIAMETER MAIN 68
C ISAV IS SAVING PARAMETER..... MAIN 69
C =0, DONOT SAVE =1 SAVE INPUT FIELD MAIN 70
C =-1 USE DATA SET #9 FOR INPUT MAIN 71
C IPHASE CONTROLS THE PHASE CONNECTIONS APPLIED TO THE FIELD MAIN 72
C = 0 NONE MAIN 73
C = 1 PLANAR CONNECTION MAIN 74
C = 2 SPHERICAL MAIN 75
C = 3 BOTH MAIN 76
C HBB IS THE BUCKET SIZE FOR OPTIMAZATION...IF A CALL TO QUAL IS MAIN 77
  DONE BEFORE OPTIMUM THE BUCKET IS SPECIFIED HERE MAIN 78
C HF IS HL/D RADIUS FOR QUALITY CALCULATION MAIN 79
C MAIN 80
C.....C MAIN 81
C MAIN 82
C MAIN 83
C NAMELIST/ OPTIM / HB, IPUT, NIND, NBIGIT, DHB MAIN 84
C HB = BUCKET SIZE FOR QUALITY OPTIMIZATION MAIN 85
C IPUT = 1 POWER WITHIN HB MAIN 86
C 2 TOTAL POWER IN BEAM MAIN 87
C 3 PEAK INTENSITY MAIN 88
C NIND IS NUMBER OF IND VARIABLES TO BE OPTIMIZED MAIN 89
C MAIN 90

```

```

C          NBIGIT = BIGGEST NUMBER OF ITERATIONS TO BE PERFORMED          MAIN          91
C          DBD IS THE BEAM DIAMETER FOR QUALITY CALC...IF CALLED TO QUAL    MAIN          92
C          EARLIER THIS IS NOT NEEDED                                       MAIN          93
C                                                                              MAIN          94
C          NAMELIST/OPT2/ IEL1, IEL2, IEL3, AMIN, XMAX, XADD                MAIN          95
C          (IEL1,IEL2,IEL3) IS THE VECTOR DESCRIBING THE POSITION OF THE     MAIN          96
C          OPTIMIZED PARAMETER...IN OPERATIONAL SPACE                       MAIN          97
C          AMIN AND XMAX ARE THE CONSTRAINTS ON THE OPTIMIZED VECTOR       MAIN          98
C          XADD IS A CONSTANT ADDED TO THE OPTIMIZED VARIABLE SUCH THAT    MAIN          99
C          ITS VALUE IS NEVER EQUAL TO ZERO                                  MAIN         100
C          THERE ARE NIND NUMBER OF CALLS TO THIS NAMELIST                 MAIN         101
C                                                                              MAIN         102
C                                                                              MAIN         103
C.....C MAIN         104
C                                                                              MAIN         105
C          NAMELIST / PARAM / MEL1,MEL2,MEL3, NPARAM, XPARAM,             MAIN         106
C          MEL1,MEL2,MEL3, NPARAM, XPARAM                                   MAIN         107
C          (MEL1,MEL2,MEL3) IS THE VECTOR DESCRIBING THE POSITION OF THE     MAIN         108
C          VARIABLE WHICH IS TO BE VARIED                                   MAIN         109
C          NPARAM,XPARAM ARE THE NUMBER OF CHANGES IN EACH VARIABLE       MAIN         110
C          XPARAM,XPARAM ARE THREE ARRAYS THAT CONTAIN THE VALUES WHICH   MAIN         111
C          ARE TO BE USED                                                  MAIN         112
C          *****IF ONLY ONE SET IS TO BE VARIED USE ONLY THE NPARAM SET, MAIN         113
C          AND SET MEL1 = 0. THE NES ARE THE INNER LOOP *****          MAIN         114
C          IF ABC ARRAY IS TO BE CHANGED AND NO CALL, AT THIS TIME TO     MAIN         115
C          AUTO(GDL), THEN SET NPARAM=0...THEN TWO VALUES CAN BE CHANGED MAIN         116
C          IF ONLY ONE IS TO BE CHANGED SET MEL1=0                        MAIN         117
C          *****ALL CALLS BETWEEN GDL AND PARAM TO QUAL,PLUT...WILL BE REPEATED MAIN         118
C          INSIDE THE PARAMETRIC LOOP                                       MAIN         119
C.....C MAIN         120
C                                                                              MAIN         121
C                                                                              MAIN         122
C          NAMELIST / THREE / PLOT3D,TITLE3,DIAM,                          MAIN         123
C          PLTISU, MPLUT, DIAISU, PSlice, NP, JFAZE, XMAG                  MAIN         124
C                                                                              MAIN         125
C                                                                              MAIN         126
C          PLOT3D = .TRUE. FOR THREE DIMENSIONAL PLOTS OF NEAR FIELD      MAIN         127
C          = .FALSE. FOR NO PLOTS                                          MAIN         128
C          TITLE3 = TITLE INFORMATION FOR THREE DIMENSIONAL PLOTS        MAIN         129
C          DIAM = DIAM OF ILLUSTRATED FIELD ON PLOTS                       MAIN         130
C                                                                              MAIN         131
C          PLTISU IS LOGICAL FOR ISOPLOTS OF FIELD                         MAIN         132
C          MPLUT IS THE RADIUS OF CIRCLE DRAWN ON ISOPLOT FOR REFERENCE    MAIN         133
C          DIAISU IS DIAMETER OF ISOPLOTS DESIRED                          MAIN         134
C          PSlice IS LOGICAL FOR SLICE PLOTS OF FIELD                      MAIN         135
C          NP = THE SLICE IN Y-DIM. PLUTTTED, IF = 0... NP = NPIS/2      MAIN         136
C          JFAZE = 0, NO PHASE PLOT FOR THIS                               MAIN         137
C          = 1, GET THE PHASE                                              MAIN         138
C                                                                              MAIN         139
C                                                                              MAIN         140
C.....C MAIN         141
C          CALL LIST80(5)                                                  CURR1          4
C          INT=0                                                            MAIN         143
C          ICNVNU=0                                                         MAIN         144
C          WL=-1.                                                            MAIN         145
C          DNX = 0.                                                          MAIN         146
C          DNY = 0.                                                          MAIN         147
C          PI=3.141592                                                       MAIN         148
C          DU 14 II=1.4                                                       MAIN         149
C          14 XSCH(II)=1.                                                    MAIN         150
C          IMNK = 1                                                           MAIN         151
C          MAINE(1) = 1                                                       MAIN         152
C          INULU=5                                                            MAIN         153
C          NP = 8.                                                            MAIN         154
C          999 HEAD(5,START)                                                  CURR1          5
C          WL = WNL                                                           CURR1          6
C          NPIS=NPNIS                                                         CURR1          7
C          DNX=DUNX                                                           CURR1          8
C          DNY=DUDNY                                                           CURR1          8
C          PLUTSU = PLUTS                                                     LNUP1          8
C          HEAD (5,12+3) TITLES                                             MAIN         156
C          12+3 FUMAT (20A+)                                                 MAIN         157

```

	IF (NL .LE. 4) GO TO 4H70	MAIN	158
	WRITE(6,150) TITLES	MAIN	159
150	FORMAT(1M1.3U(1M50U)/1X.1MU.8BA.1MU/1X.1MU.4A.2UA4.4A.1MU/	MAIN	160
	X 1A.1MU.8HX.1MU/1X.3U(1M50U)///)	MAIN	161
	NPY = NPFS	MAIN	162
	IF (SYMTRC) NPY = NPY/2	MAIN	163
	NUB = NPFS * NPY	MAIN	164
	NJKY = 0	MAIN	165
	ABC(1,2,1) = ONX	MAIN	166
	ABC(2,2,1) = ONY	MAIN	167
	IMK = IMK + 1	MAIN	168
	MAINE(1IMK) = NCALL	MAIN	169
	GO TO (999,100,200,300,400,500),NCALL	MAIN	170
C	MAIN	171
C	TRANSFER CONTROL TO GDL	MAIN	172
100	IF (RESINT .OH. 18.E4.0) GO TO 3	MAIN	173
	DX=DCAL/NPFS	MAIN	174
	X(1)=-DCAL/2.*DX/2.	MAIN	175
	DO 2 I=2,NPFS	MAIN	176
2	X(I)=X(I-1)*DX	MAIN	177
	DO 4 I=1,NUB	MAIN	178
9	CU(I) = CMPLX(AMPGES,0.)	MAIN	179
	IF (PHIRAD.E4.0.)GO TO 71	MAIN	180
	MUFACT=-PI/(18.*PHIRAD)	MAIN	181
	DO 72 J=1,NPY	MAIN	182
	J1=(J-1)*NPFS	MAIN	183
	YSU = X(J)*.02	MAIN	184
	DO 72 I=1,NPFS	MAIN	185
	KKK=J1+I	MAIN	186
	KKK2 = 2 * KKK	MAIN	187
	KKK2M1 = KKK2 - 1	MAIN	188
	PHI = MUFACT * (X(I)*.02+YSU)	MAIN	189
	SINP = SIN(PHI)	MAIN	190
	CUSP = COS(PHI)	MAIN	191
	CUMS = CUM(KKK2M1)	MAIN	192
	CUM(KKK2M1) = CUMS*CUSP - CUM(KKK2)*SINP	MAIN	193
72	CUM(KKK2) = CUMS*SINP + CUM(KKK2)*CUSP	MAIN	194
71	IF(DGAUSS.E4.0.) GO TO 50	MAIN	195
	SIGMA=DGAUSS*.02/4.0	MAIN	196
	DO 51 J=1,NPY	MAIN	197
	NHUM=(J-1)*NPFS	MAIN	198
	YSU = X(J)*.02	MAIN	199
	DO 51 I=1,NPFS	MAIN	200
	NCNT=NHUM+I	MAIN	201
	CU(NCNT)=CU(NCNT)*EXP(-(X(I)*.02+YSU)/SIGMA)	MAIN	202
51	CONTINUE	MAIN	203
	WRITE(6,52)DGAUSS,AMPGES	MAIN	204
52	FORMAT(15MU GAUSSIAN AMPLITUDE DISTRIBUTION HAS BEEN FORMED WITH	MAIN	205
	X A 1/E AMPLITUDE AT DIAMETER=.F10.2/18M PEAK AMPLITUDE=.612.5/)	MAIN	206
50	CONTINUE	MAIN	207
	NIT = 0	MAIN	208
	GO TO 4	MAIN	209
3	IF (18.E4.0) GO TO 4	MAIN	210
	HEAD(18) (CU(I),I=1,NUB),X,UM1,UM2,NIT	MAIN	211
	NEWIND 18	MAIN	212
	* IF (1M.E4.0) GO TO 999	MAIN	213
	IF (1M.E4.1NULD.OH.1N.E4.5.) GO TO 5	MAIN	214
	WRITE (6,6) IN	MAIN	215
6	FORMAT (27M1 THE INPUT DATA ON SET # .12.21M FOR THIS CALL TO GDL	MAIN	216
	X/)	MAIN	217
	CALL LISTEN(IN)	MAIN	218
	INULD=IN	MAIN	219
5	IF (11NUM .GE. 0) NIT = 11NUM	MAIN	220
	CALL GDL(IN,RESINT,ABC,NIT,[B.0])	MAIN	221
	MULD = IMK	MAIN	222
	CALUL = .FALSE.	MAIN	223
	GO TO 999	MAIN	224
C	MAIN	225
C	TRANSFER CONTROL TO QUAL	MAIN	226
200	HEAD(5,4L01)	MAIN	227
	HEAD (5,12*3) TITLE	MAIN	228
	CALQL = .TRUE.	MAIN	229

210	CALL QUAL (IMHASE,ISAV,IULI,ITITLE,HRB,AS,UB,MF)	MAIN	230
	GO TO 997	MAIN	231
C	MAIN	232
C	TRANSFER CONTROL TO PLOTTING ROUTINES	MAIN	233
300	HEAD(5,IMHED)	MAIN	234
	IF(XMAG.EQ.1.)GO TO 310	MAIN	235
	DU 377 IMG=1,NPTS	MAIN	236
377	X(IMG)=X(IMG)*XMAG	MAIN	237
	DU 378 IMG=1,NUB	MAIN	238
378	CU(IMG)=CU(IMG)/XMAG	MAIN	239
310	IF (PLOT3D) CALL NEAR(DIAM,ITITLE)	MAIN	240
	IF (PLTISO) CALL ISOS(TITLE,IRPLUT,UIAISO)	MAIN	241
	IF (PSLICE) CALL PHETYP(INP,ITITLE,JPFAZE)	MAIN	242
	GO TO 997	MAIN	243
C	MAIN	244
C	PERFORM GRADIENT SEARCH OPTIMIZATION	MAIN	245
400	DO 8 II=1,NIND	MAIN	246
	HEAD(5,UPT2)	MAIN	247
	IUP(1,II) = IEL1	MAIN	248
	IUP(2,II) = IEL2	MAIN	249
	IUP(3,II) = IEL3	MAIN	250
	XLOW(II) = XMIN	MAIN	251
	XUP(II) = XMAX	MAIN	252
	XOPADD(II) = XAOD	MAIN	253
8	XUP(II) = ABC(IUP(1,II),IUP(2,II),IUP(3,II))*XOPADD(II)	MAIN	254
410	IF (.NOT.CALQL) CALL QUAL(0,0,0,ITITLE,HRB,AS,UBH)	MAIN	255
32	QQQ = 1./ AS(IPUT)	MAIN	256
	CALL CNSTRN (XOP,XLOW,XUP,NIND,QQQ,QQQQ)	MAIN	257
	CALL DAVIUM (QQQQ,XOP,NIND,ICNVRG,NBIBIT,.002,0.,XSCH)	MAIN	258
	GO TO (919,918),ICNVRG	MAIN	259
	DU 13 II=1,NIND	MAIN	260
13	ABC(IUP(1,II),IUP(2,II),IUP(3,II)) = XUP(II) -XOPADD(II)	MAIN	261
C	CALL AUTO(ABC,IB)	MAIN	262
	CALL GOL(IN,HESTHT,ABC,NIT,IB,1)	MAIN	263
	IF (MULU.EQ.1MRK-1) GO TO 410	MAIN	264
	NURY = MULU	MAIN	265
	GO TO 997	MAIN	266
919	WRITE (6,23) (ABC(IUP(1,II),IUP(2,II),IUP(3,II)),II=1,NIND), AS	MAIN	267
23	FORMAT (B2M1 OPTIMIZATION ROUTINE HAS CHOSEN THE FOLLOWING PARA	MAIN	268
	ETERS...AND THIS MAXIMUM /5P12.577)	MAIN	269
	GO TO 999	MAIN	270
918	WRITE (6,24)	MAIN	271
24	FORMAT (B2M1 OPTIMUM ROUTINE HAS EXCEEDED MAX # OF ITERATIONS)	MAIN	272
	GO TO 999	MAIN	273
C	MAIN	274
C	PERFORM PARAMETRIC STUDY	MAIN	275
500	HEAD(5,PARAM)	MAIN	276
	JJ1 = 0	MAIN	277
	IF (NPARAM.NE.0) GO TO 510	MAIN	278
	IMRK = IMRK - 1	MAIN	279
	ABC(MEL1,MEL2,MEL3) = XMPARA(1)	MAIN	280
	IF (MPARA.EQ.0) ABC(MEL1,MEL2,MEL3) = XMPARA(1)	MAIN	281
	GO TO 999	MAIN	282
510	JJ2 = 0	MAIN	283
	IF (MEL1.EQ.0) MPARAM = 1	MAIN	284
	JJ1 = JJ1 + 1	MAIN	285
	IF (JJ1.GT.MPARAM) GO TO 999	MAIN	286
	IF (MEL1 .NE. 0)	MAIN	287
	X ABC(MEL1,MEL2,MEL3) = XMPARA(JJ1)	MAIN	288
520	JJ2 = JJ2 + 1	MAIN	289
	IF (JJ2 .GT. NPARAM) GO TO 510	MAIN	290
	ABC(MEL1,MEL2,MEL3) = XMPARA(JJ2)	MAIN	291
	CALL GOL(IN,HESTHT,ABC,NIT,IB,1)	MAIN	292
C	CALL AUTO(ABC,IB)	MAIN	293
	NURY = MULU	MAIN	294
	GO TO 997	MAIN	295
997	NURY = NURY + 1	MAIN	296
	NNN = MAINE(NURY)	MAIN	297
	GO TO (999,100,210,310,410,520), NNN	MAIN	298
9876	STOP	MAIN	299
	END	MAIN	300

2. SUBROUTINE LIST80

Calls: N/A

Called by: MAIN

Subroutine LIST80 is called by the executive routine MAIN to list data input to the SOQ code. The LIST80 flow chart (Fig. 13) appears on page 45.

After control is passed to LIST80, header information is printed. The input unit is read and a counter, KARD, is incremented for each record read. The input data is reformatted and printed on the line printer. When an end-of-file is received from the input unit, it is backspaced K records and control is returned to MAIN.

Arguments

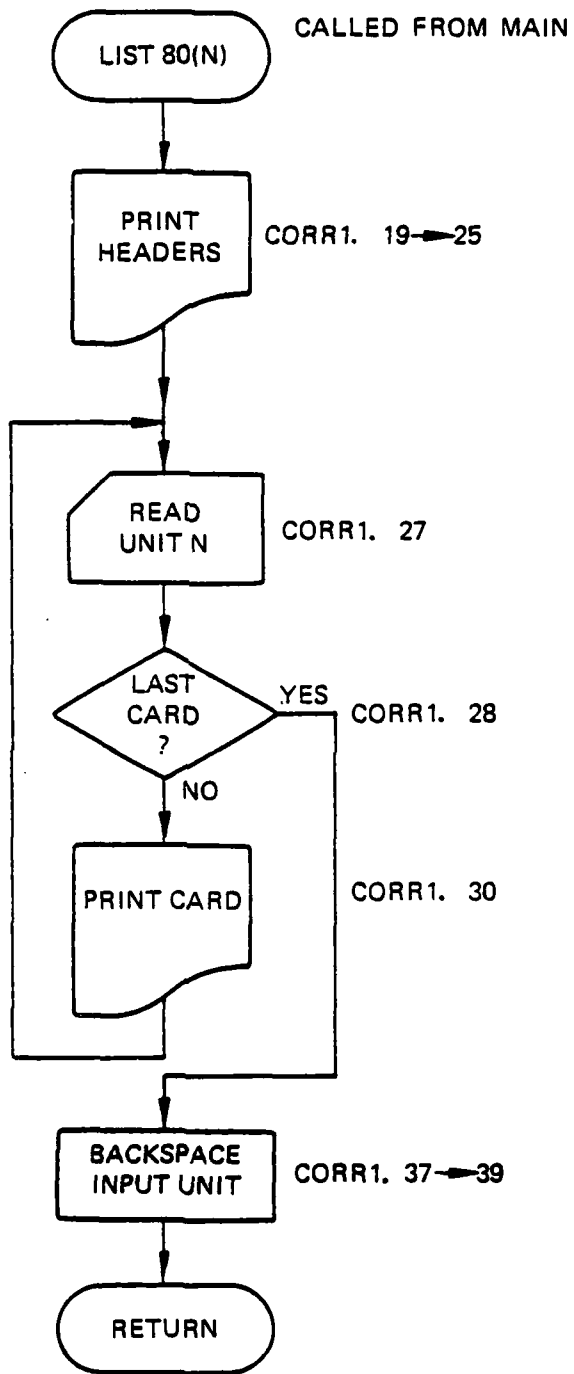
K Unit number on which input is read (usually 5).

Relevant Variables

C Card inputs read and printed as read.

SUBROUTINE LIST80 76/176 OPT=1 FIN 4.6+452 04/27/79 12.25.47

C	SUBROUTINE LIST80(K)	CUMRI	14
	THIS ROUTINE WRITES NAMELIST INPUT HOPEFULLY	CUMRI	15
	DIMENSION C(20)	CUMRI	16
	WRITE(6,35)	CUMRI	17
	KARD = 1	CUMRI	18
	30 WRITE(6,20)	CUMRI	19
	20 FORMAT(4(I),1X,52(1M*),12M*CARD INPUTS, 52(1M*))	CUMRI	20
	WRITE(6,40)	CUMRI	21
	40 FORMAT(/)	CUMRI	22
	=5M*CARD,15A,10(1M),10(1M2),10(1M3),10(1M4),10(1M5),	CUMRI	23
	=10(1M6),10(1M7),1M6,7M3COLUMN,4X,8(10M1234567890),5A,	CUMRI	24
	=5M*CARD,/))	CUMRI	25
	DO 25 J = 1,5	CUMRI	26
	1 READ(K,5) C	CUMRI	27
	IF (EOF(K).NE.0.0)GO TO 15	CUMRI	28
	5 FORMAT(20A)	CUMRI	29
	WRITE(6,10) C,KARD	CUMRI	30
	10 FORMAT(10A,20A,18)	CUMRI	31
	KARD = KARD + 1	CUMRI	32
	25 CONTINUE	CUMRI	33
	WRITE(6,40)	CUMRI	34
	WRITE(6,35)	CUMRI	35
	GO TO 30	CUMRI	36
	15 IBACK = KARD - 1	CUMRI	37
	DO 45 I = 1,IBACK	CUMRI	38
	45 BACKSPACE K	CUMRI	39
	WRITE(6,40)	CUMRI	40
	WRITE(6,35)	CUMRI	41
	35 FORMAT(1M)	CUMRI	42
	RETURN	DUMMYS	27
	END	DUMMYS	28



FD 162893

Figure 13. Subroutine LIST80 flow chart.

3. SUBROUTINE AEROW

Subroutine AEROW is used to apply a random phase variation to the complex field. Figure 14 shows the subroutine AEROW flow chart.

AERO is entered with the complex field array real coefficients, CUR, and with the number of points in x and y.

SIGMAM is a constant established by previous aerowindow work. It is later multiplied by the random number returned from the RANDU call to give the proper random phase range for an aerowindow.

Inside the DO LOOP, the random phase is obtained and the sine and cosine of the negative of this phase is taken. A negative number is required to yield a diverging phase impact.

The complex field, CU, is represented by a complex number, $a + ib$, whereas the CUR variables represent the real coefficients alone.

$$CU(1) \begin{cases} CUR(1) = a \\ CUR(2) = b \end{cases} \quad a + ib \quad (11)$$

The random phase is applied by:

$$\overbrace{(a + ib)}^{CU} * e^{i\phi} \quad (12)$$

$$(a + ib) (\cos \phi + i \sin \phi) \quad (13)$$

$$\left. \begin{array}{l} a \cos \phi - b \sin \phi \rightarrow CUR(1) \\ b \cos \phi + a \sin \phi \rightarrow CUR(2) \end{array} \right\} CU(1) \quad (14)$$

Argument List

CUR	Complex field array
NPTS	Number of x points
NPY	Number of y points

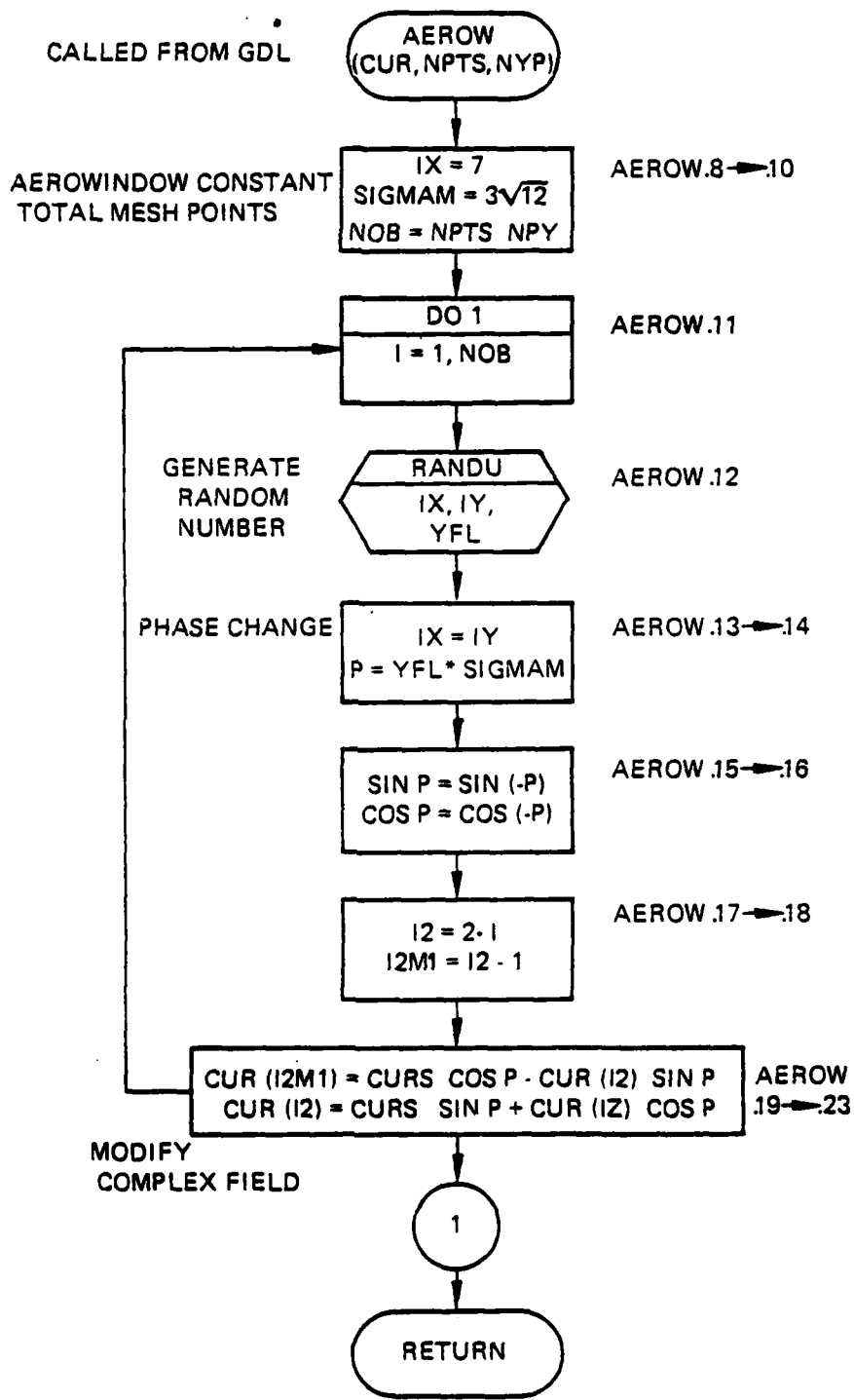


Figure 14. Subroutine AEROW flow chart.

Relevant Variables

CURS Odd number members of field CUR
P Phase change
SIGMAM Aerodynamic window constant = $0.3\sqrt{2}$
YFL Random number generated by "RANDU"

SUBROUTINE AEROW 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE AEROW (CUR,NPTS,NPY)
C AERODYNAMIC WINDOW MODEL
C THIS ROUTINE APPLIES A RANDOM PHASE VARIATION TO THE COMPLEX
C FIELD
LEVEL 2: CUR,NPTS,NPY
DIMENSION CUR(1)
IX = 1
SIGMAM = 0.300 * SQRT(12.)
NUB = NPTS*NPY
DO 1 I = 1,NUB
CALL RANDU (IX,IY,YFL)
IX = IY
P = YFL * SIGMAM
SINP = SIN(-P)
COSP = COS(-P)
I2 = 2*I
I2M1 = I2 - 1
CURS = CUR(I2M1)
CUR(I2M1) = CURS*COSP - CUR(I2)*SINP
1 CUR(I2) = CURS*SINP + CUR(I2)*COSP
C 1 CUR(I) = CUR(I) * CEXP(CMPLX(I0,-P))
RETURN
END
AEROW 2
AEROW 3
AEROW 4
AEROW 5
AEROW 6
AEROW 7
AEROW 8
AEROW 9
AEROW 10
AEROW 11
AEROW 12
AEROW 13
AEROW 14
AEROW 15
AEROW 16
AEROW 17
AEROW 18
AEROW 19
AEROW 20
AEROW 21
AEROW 22
AEROW 23
AEROW 24

```

4. SUBROUTINE RANDU

Subroutine called by AEROW returns rectangularly distributed random numbers in the range 0 to 1 in the variable YFL. Figure 15 shows the RANDU flow chart.

SUBROUTINE RANDU 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE RANDU (IX,IY,YFL)
C RANDOM NUMBER GENERATOR
C THIS ROUTINE SUPPLIES THE RANDOM NUMBERS TO AEROW
IX = IX*899
IF (IY) 5,5,6
5 IY = IY*2147483647 * 1
6 YFL = IY
YFL = YFL/2147483647.
RETURN
END
RANDU 2
RANDU 3
RANDU 4
RANDU 5
RANDU 6
RANDU 7
RANDU 8
RANDU 9
RANDU 10
RANDU 11

```

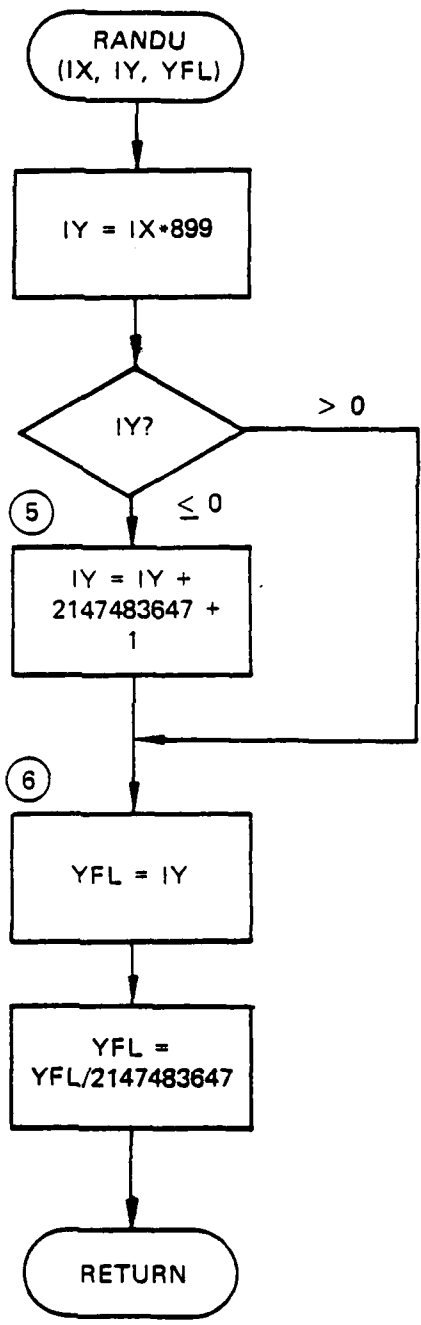


Figure 15. Subroutine RANDU flow chart.

5. SUBROUTINE APRTR

Called by: MIRROR, GDL

Calls: N/A

a. Purpose -- Subroutine APRTR applies an aperture, either circular or rectangular (Fig. 16), with or without a central obscuration, to the complex field. It also determines the value and position of maximum intensity on the aperture plate. Figure 17 shows the APRTR flow chart.

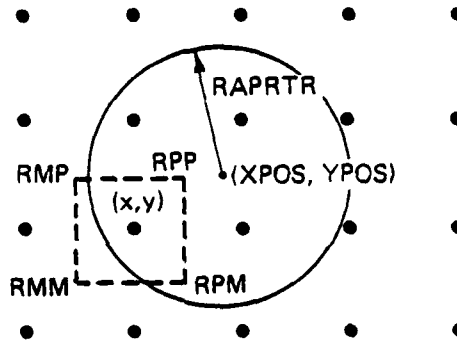


Figure 16. Subroutine APRTR nomenclature.

APRTR is entered with the inner and outer obscuration dimensions along with the coordinates of the aperture.

A test is made to see if the aperture is rectangular or circular. The appropriate boundary parameters are computed. Each point in the complex field is checked to see if it will pass through the clear aperture. If so, it is left alone. If not, it is zeroed out after it has been checked to determine if it is the location of maximum intensity on the aperture plate.

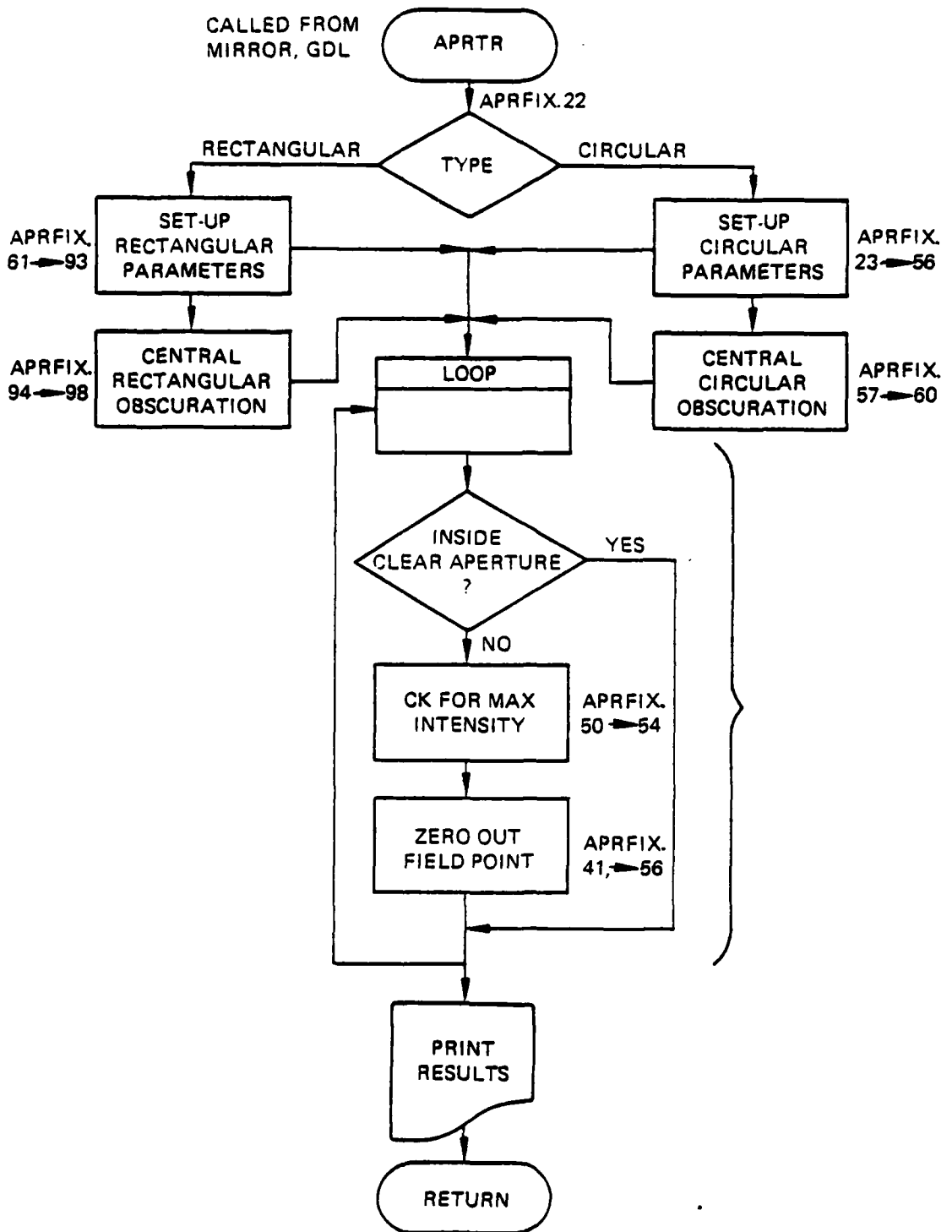


Figure 17. Subroutine APRTR flow chart.

The transmission function is

$$t(x, y) = \begin{cases} \text{RDISK} \leq \sqrt{(x-x_{\text{pos}})^2 + (y-y_{\text{pos}})^2} \leq \text{RAPRTR} \\ 0 \text{ otherwise} \end{cases} \quad (15)$$

b. Relevant formalism

$$\text{RPP} = \left(|x| + \frac{dx}{2} \right)^2 + \left(|y| + \frac{dy}{2} \right)^2 \quad (16)$$

$$\text{RMM} = \left(|x| - \frac{dx}{2} \right)^2 + \left(|y| - \frac{dy}{2} \right)^2 \quad (17)$$

$$\text{RMP} = \left(|x| - \frac{dx}{2} \right)^2 + \left(|y| + \frac{dy}{2} \right)^2 \quad (18)$$

$$\text{RPM} = \left(|x| + \frac{dx}{2} \right)^2 + \left(|y| - \frac{dy}{2} \right)^2 \quad (19)$$

These four locations represent an area surrounding the particular point of interest as shown in Figure 16. For each of these sets of points the locations of the aperture and obscuration are checked. If all the four points impinge on an aperture or central obscuration, then the intensity at that location is computed and checked for maximum value, then the field is zeroed out (by the impingement).

$$\text{Int} = (\text{ReCu})^2 + (\text{ImCu})^2 \quad (20)$$

$$\text{Max Int} = \text{AMAX} (\text{Int}, \text{Max Int}) \quad (21)$$

$$\text{PER} = 0 \quad (22)$$

$$\text{Cu} = \text{CU} \times \text{PER} \quad (23)$$

If all four points lie within the clear aperture, the field is unchanged.

$$\text{PER} = 1 \quad (24)$$

$$\text{CU} = \text{CU} \times \text{PER} \quad (25)$$

If the four points encompass an aperture edge, then the intensity is prorated on a percentage basis and transmitted.

$$\text{PER} = (\text{RAD} - \text{RMIN}) / \text{RMAX} - \text{RMIN} \quad (26)$$

$$\text{CU} = \text{CU} \times \text{PER} \quad (27)$$

where

$$\text{RMAX} = \text{MAX of (RPP, RMM, RMP, RPM)} \quad (28)$$

$$\text{RMIN} = \text{MIN of (RPP, RMM, RMP, RPM)} \quad (29)$$

$$\text{RAD} = \text{Radius (or x or y dimension) at aperture edge} \quad (30)$$

Argument List

RAPRTR Radius of circular aperture (cm) or x-dimension (half width) of rectangular aperture (cm)

RDISK Radius of central obscuration of a circular aperture (cm); or x-dimension (half width) of a rectangular central obscuration

XPOS x location of aperture center with respect to optic center-line (cm)

YPOS y location of aperture center with respect to optic center-line (cm)

YAPRTR y dimension (half height) of rectangular aperture (cm)

YDISK y dimension (half height) of a rectangular central obscuration (cm).

Relevant Variables

A Half width of rectangular aperture (cm)
AINT Intensity (W/cm^2)
AINTMX Maximum intensity (W/cm^2)
B Half height of rectangular aperture (cm)
DX x distance between points in the mesh (cm)
DY y distance between points in the mesh (cm)
RAD = RAPRTR, aperture radius (cm)
X x location adjusted for centerline difference and accumulated dx (cm)
XAR (N) x or y position of N (cm)
Y y location adjusted for centerline difference and accumulated dy (cm).

Commons Modified

/MELT/

Array modified CU(I) @ APRFIX.56,93.

SUBROUTINE APRTR 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE APRTR (RAPRTR, RDISK, XPOS, YPOS, YAPRTR, YDISK)
C
C   USUBROUTINE APRTR (RAPRTR, RDISK, XPOS, YPOS, YAPRTR, YDISK)
C   APERTURE MODEL
C   THIS ROUTINE APPLIES A CIRCULAR APERTURE WITH RADIUS = RAPRTR
C   AND A CENTRAL OBSCURATION WITH RADIUS = RDISK CENTERED ABOUT
C   XPOS, YPOS.
C   C ( MODIFIED 4/8/76 P. ADAMSKI FOR RECTANGULAR APERTURE
C   C OF WIDTH = 2*YAPRTR, HEIGHT = 2*YAPRTR AND A CENTRAL
C   C ) OBSCURATION OF WIDTH = 2*RDISK, HEIGHT = 2*YDISK
C   C ) CAPABILITY FOR FINDING VALUE AND POSITION OF MAX INTENSITY ON
C   C ) APERTURE PLATE ADDED 4/12/76 PAA.
C   LEVEL 2, CU
COMMON/MELT/ CU(1038), CFIL(1032), XAR(128), NL, NPTS, NPY, UMX, UMY
COMMON/INT/ WNO, WNEG, HAPTR
COMPLEX CU, CFIL
RU(XA,YY,IX,IY)=SQRT((ABS(XA)+IX*UX/2.)**2+(ABS(YY)+IY*UY/2.)**2)
UA=XAR(2)-XAR(1)
UY=UA
IIN=0
C ( (
INTCR=0
AINTMX=0.
IF(YAPRTR.NE.0..OR.YDISK.NE.0.) GO TO 18U
C ( ) ***** CIRCULAR APERTURE *****
WRITE(6,1000) RAPRTR,RDISK
IF(RAPRTR.EQ.0.0) GO TO 18U
RAD=RAPRTR
99 DO 101 IIX=1,NPTS
X=XAR(IIX)+UMX-XPOS
DO 101 IY=1,NPY
Y=XAR(IY)+UMY-YPOS
APNFI X 1
APNFI X 2
APNFI X 3
APNFI X 4
APNFI X 5
APNFI X 6
APNFI X 7
APNFI X 8
APNFI X 9
APNFI X 10
APNFI X 11
APNFI X 12
APNFI X 13
APNFI X 14
APNFI X 15
APNFI X 16
APNFI X 17
APNFI X 18
APNFI X 19
APNFI X 20
APNFI X 21
APNFI X 22
APNFI X 23
SWAPN 1
APNFI X 24
APNFI X 25
APNFI X 26
APNFI X 27
APNFI X 28
APNFI X 29

```


C (R =SQRT(X**2+Y**2)	APNFIX	30
	IF (H.GE.HAPNTH) INTCK=1	APNFIX	31
C (HPP=HU(A,Y,1,1)	APNFIX	32
	HMM=HD(A,Y,-1,-1)	APNFIX	33
	HMP=HO(X,Y,-1,1)	APNFIX	34
	HPM=HU(X,Y,1,-1)	APNFIX	35
	PER=1.	APNFIX	36
	HMAX=AMAX1(HPP,HMM,HMP,HPM)	APNFIX	37
	IF (HMAX.LE.HAU) GO TO 100	APNFIX	38
	PER=U.	APNFIX	39
	HMIN=AMIN1(HPP,HMM,HMP,HPM)	APNFIX	40
	IF (HMIN.GE.HAU) GO TO 100	APNFIX	41
	PER=(HAU-HMIN)/(HMAX-HMIN)	APNFIX	42
100	IF (IIN.EQ.1) PER=1.-PER	APNFIX	43
	NNN = IIX*(IYY-1)*NPTS	APNFIX	44
C (APNFIX	45
	IF(INTCK.EQ.0) GO TO 101	APNFIX	46
	INTCK=0	APNFIX	47
	AINT=MEAL(CU(NNN))**2 + AIMA(CU(NNN))**2	APNFIX	48
	AINTM=AMAX1(AINT,AINTM)	APNFIX	49
	IF (AINT.NE.AINTM) GO TO 101	APNFIX	50
	AINTM=A	APNFIX	51
	YINTM=B	APNFIX	52
C (APNFIX	53
101	CU(NNN) = CU(NNN) * SQRT(PER)	APNFIX	54
100	IF (HUISK.EQ.0.OR.IIN.EQ.1) GO TO 300	APNFIX	55
	IIN=1	APNFIX	56
	HAU=HUISK	APNFIX	57
	GO TO 30	APNFIX	58
C ((***** RECTANGULAR APERTURE *****	APNFIX	59
100	CONTINUE	SUAPH	2
	HU=2.*HAPNTH	SUAPH	3
	HU=2.*YAPNTH	SUAPH	4
	HI=2.*HUISK	SUAPH	5
	HI=2.*YUISK	SUAPH	6
	WRITE(6,1001) HU,HU,HI,HI	SUAPH	7
1000	FORMAT(/' CIRCULAR APERTURE APPLIED'//' OUTSIDE RADIUS =',G8.3	SUAPH	8
	X,/' INSIDE RADIUS =',G8.3//)	SUAPH	9
1001	FORMAT(/' RECTANGULAR APERTURE APPLIED'//' OUTSIDE DIMENSIONS	SUAPH	10
	NAME =',G8.3,' HIGH BY ',G8.3,' WIDE'//' INSIDE DIMENSIONS ARE ',	SUAPH	11
	X G8.3,' HIGH BY ',G8.3,' WIDE'//)	SUAPH	12
	IF(HAPNTH.EQ.0.0) GO TO 200	SUAPH	13
	A =HAPNTH	APNFIX	63
	B =YAPNTH	APNFIX	64
199	DO 201 IIX=1,NPTS	APNFIX	65
	X=XAH(IIX)*DHX-XPOS	APNFIX	66
	DO 201 IYY=1,NPY	APNFIX	67
	Y=YAH(IYY)*DHY-YPOS	APNFIX	68
C (APNFIX	69
	IF (ABS(X).GE.HAPNTH.OR.ABS(Y).GE.YAPNTH) INTCK=1	APNFIX	70
C (APNFIX	71
	XMIN = ABS(X)-UX/2	APNFIX	72
	XMAX = ABS(X)+UX/2	APNFIX	73
	YMIN = ABS(Y)-UY/2	APNFIX	74
	YMAX = ABS(Y)+UY/2	APNFIX	75
	PER=0.	APNFIX	76
	IF (XMIN.GE.A.OR.YMIN.GE.B) GO TO 200	APNFIX	77
	PER=1.	APNFIX	78
	IF (XMAX.LE.A.AND.YMAX.LE.B) GO TO 200	APNFIX	79
	IF (XMAX.GE.A) PER=(A-XMIN)/UA	APNFIX	80
	IF (YMAX.GE.B) PER = PER * (B-YMIN)/UY	APNFIX	81
200	IF (IIN.EQ.1) PER=1.-PER	APNFIX	82
	NNN = IIX*(IYY-1)*NPTS	APNFIX	83
C (APNFIX	84
	IF(INTCK.EQ.0) GO TO 201	APNFIX	85
	INTCK=0	APNFIX	86
	AINT=MEAL(CU(NNN))**2 + AIMA(CU(NNN))**2	APNFIX	87
	AINTM=AMAX1(AINT,AINTM)	APNFIX	88
	IF (AINT.NE.AINTM) GO TO 201	APNFIX	89
	AINTM=A	APNFIX	90
	YINTM=B		

C)		APNFIX	91
201	CU(NNN) = CU(NNN) * SQRT(PEN)	APNFIX	92
200	IF (WDISK.EQ.0..OR.IIN.EQ.1) GO TO 300	APNFIX	93
	IIN=1	APNFIX	94
	A = WDISK	APNFIX	95
	B = YDISK	APNFIX	96
	GO TO 199	APNFIX	97
C (APNFIX	98
300	FAF=1.	APNFIX	99
	IF (NNEG.EQ.1..OR.NNEG.EQ.2) FAF=1./NNEG	APNFIX	100
	AINTMA=AINTMA*FAF	APNFIX	101
		APNFIX	102
	WRITE(6,310) AINTMA,XINTMA,YINTMA	APNFIX	103
310	FORMAT(' THE MAX INTENSITY ON APERTURE PLATE IS (MAX= %.G13.5/	APNFIX	104
	1' AND IS LOCATED AT X= %.F13.5, Y= %.F13.5)	APNFIX	105
	RETURN	APNFIX	106
C))		APNFIX	107
	END	APNFIX	108

6. SUBROUTINE BLUMIT

a. Purpose -- In the interstage duct, phase perturbation can be induced in the beam due to transient thermal blooming. This effect is suppressed by a sonic purge flow using the transverse thermal blooming routine. The BLUMIT routine models this residual sonic purge flow thermal blooming in the interstage duct. Figure 18 shows the subroutine BLUMIT organization.

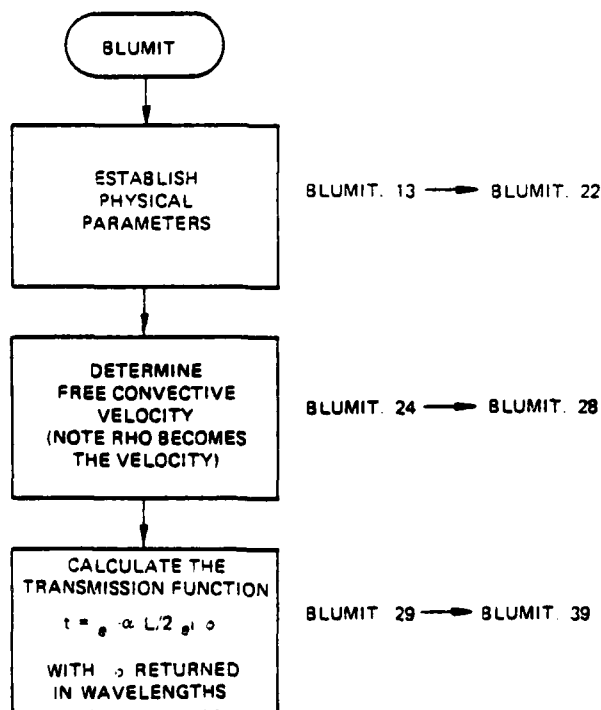


Figure 18. Subroutine BLUMIT organization.

b. Formalism -- As the beam propagates through the sonic purge flow, it is continuously distorted by that flow. Under the assumption that this distortion has a perturbative effect on the beam, the integrated effect of any thermal blooming can be approximated by a finite number of discrete steps in the following manner:

Assume each step is of length DL. The distortion is applied by propagating a length DL/2 to the center of the cell, then applying the thermal blooming transmission function. The beam is then propagated through the remaining DL/2 to the edge of the cell. The nonlinear blooming transmission function $t(x,y,\Delta L,I(x,y))$ is

$$t(x,y,\Delta L,I(x,y)) = e^{-\alpha\Delta L/2} e^{i\Delta\phi} \quad (31)$$

where, α is the absorptivity of the medium. $\Delta\phi$ is written

$$\Delta\phi = \frac{2\pi}{\lambda} \frac{dn}{dT} \int_0^{\Delta L} dz' \delta T(x,y,z') \quad (32)$$

This can be rewritten using the equation of state for an ideal gas ($P = RT_0/M$) and the Gladstone-Dale relationship. Assuming constant pressure, the expression of $\Delta\phi$ becomes

$$\Delta\phi = \frac{2\pi}{\lambda} \left(-\frac{\rho C_{G-D}}{T} \right) \int_0^{\Delta L} dz' \delta T(x,y,z') \quad (33)$$

where δT represents the temperature variation in the flow due to heating by the beam. For transverse blooming, δT can be written

$$\delta T = \frac{\alpha}{\rho C_p v_T} \int_{-\infty}^x dx' I(x',y,z) \quad (34)$$

In the above expression, the flow is assumed to be from the negative X direction with speed v_T .

This effect is activated in subroutine CAVITY by setting NGTYPE=2. The duct is then treated as if it were another cavity, the gain/phase transmission function being that of transverse thermal blooming. It is updated by subroutine REGAIN.

Since the only mathematical difference between transverse and free convective is in the velocity, this routine can also handle free convection blooming with

$$v_{fc} = \left(\frac{2\alpha P(z)g}{\rho C_p T} \right)^{1/3} \quad (35)$$

c. Fortran

Argument list

P = Intensity array. It returns as the phase change in wavelengths due to blooming.

G = Gain array. Intensity loss due to blooming.

NCV = Cavity number

WL = Wavelength

Commons modified - None

Subroutines called - None

The subroutine BLUMIT computer printout follows.

SUBROUTINE BLUMIT 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE BLUMIT(P,G,NCV,WL)
C INTERSTAGE DUCT THERMAL BLOOMING MODEL
C THIS ROUTINE CALCULATES THE COMPLEX TRANSMISSION FUNCTION OF THE
C ALL INTERSTAGE DUCT AS A FUNCTION OF THE POWER DENSITIES WITHIN
C THE DUCT
C THIS CODE IS PHELIMINARY
C LEVEL 2: P,G,XC,WL
COMMON/CAV2/AC(5),YC(5),ZC(5),NX(5),NY(5),NS(5),XMC(5),YMC(5),
1 NN(20),S2(196,5),TV1(5),TV2(5),TV3(5),TVN2(5),ISCAV(5),S3(35),
2 TITLE(20),AVG(5),NSYM
DIMENSION P(1),G(1)
ANGL = I3CAV(NCV)
ALFA = TV1(NCV)
CP = TV2(NCV)
RMU = TV3(NCV)
I = TVN2(NCV)
DELZ = ZC(NCV)/NS(NCV)
BLUMIT 2
BLUMIT 3
BLUMIT 4
BLUMIT 5
BLUMIT 6
BLUMIT 7
BLUMIT 8
BLUMIT 9
BLUMIT 10
BLUMIT 11
BLUMIT 12
BLUMIT 13
BLUMIT 14
BLUMIT 15
BLUMIT 16
BLUMIT 17
BLUMIT 18

```

NAA = NX(NCV)	BLUMIT	19
NYA = NY(NCV)/(NSYM+1)	BLUMIT	20
MUF=NXA*NYA	BLUMIT	21
DELX = AC(NCV)/NXA	BLUMIT	22
IF(HMU.GT.1.) GO TO 10	BLUMIT	23
SUM = 0.	BLUMIT	24
DO 12 I=1,MUF	BLUMIT	25
12 SUM = SUM+P(I)	BLUMIT	26
SUM = SUM*DELX*YC(NCV)/NY(NCV)	BLUMIT	27
RMU = (980.665*SUM*ALFA/(HMU*CP*T))**(.1./J.)	BLUMIT	28
10 CAP = .23*ALFA*DELZ*DELX/(CP+HMU)	BLUMIT	29
CAP2 = EXP(-ALFA*DELZ/2.)	BLUMIT	30
IB = +1	BLUMIT	31
IF(ANGL.GE.90.) IB=-1	BLUMIT	32
DO 20 J=1,NYA	BLUMIT	33
SUM = 0.	BLUMIT	34
DO 20 I=1,NXA	BLUMIT	35
IX = (1+NXA)*((1-IB)/2+IB*I + (J-1)*NXA	BLUMIT	36
SUM = SUM+P(IX)	BLUMIT	37
P(IX) = SUM*CAP	BLUMIT	38
20 G(I+(J-1)*NXA) = CAP2	BLUMIT	39
RETURN	BLUMIT	40
END	BLUMIT	41

7. SUBROUTINE CAVITY

a. Purpose -- The CAVITY routine models the interaction of a GDL cavity and the complex optical field. As the simulated field is propagated through the cavity, it interacts with the flowing medium. As a result, both the intensity and phase of the beam are modified through the CAVITY routine. Figure 19 shows the subroutine CAVITY organization.

b. Formalism -- As the beam is propagated through the cavity, its intensity and phase are continuously updated. The beam's amplitude and phase are amplified and redirected by the medium-induced gain and phase change. This medium-beam interaction results in an integrated effect. It is assumed in CAVITY that the total effect can be approximated by a finite sum of N terms in the following manner: The total cavity length Z is divided into N steps, each $Z/N = \Delta L$ in length. In each segment, the interaction of the field with the medium is approximated by vacuum propagation through half of the segment, $(\Delta L/2)$, followed by the application of a field dependent transmission function of the form

$$t(x,y,I) = \varepsilon^{\Delta L} \left[\left(g(x,y,I)/2 + i \frac{2\pi}{\lambda} (\Delta n(x,y,I)) \right) \right] \quad (36)$$

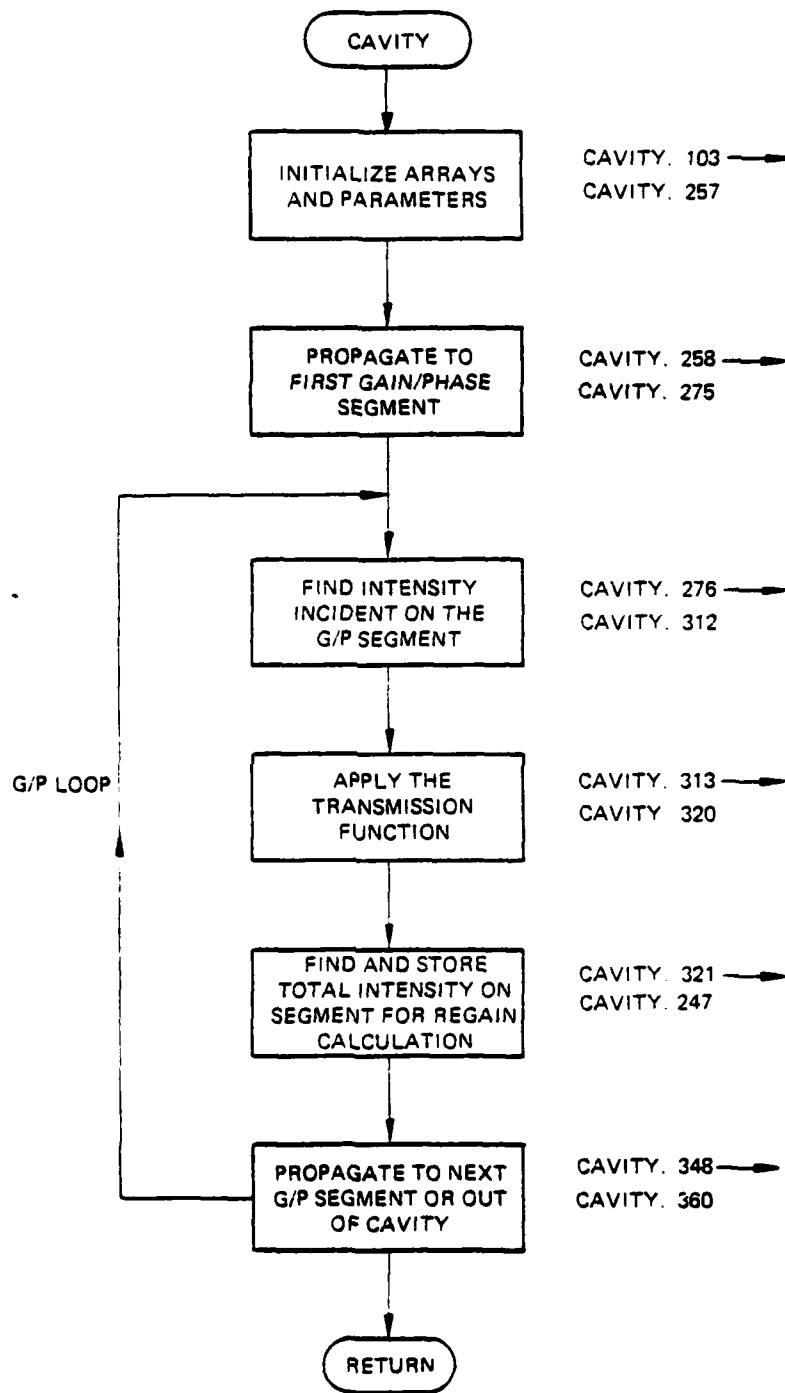


Figure 19. Subroutine CAVITY organization.

The gain coefficient g and refractive index Δn are calculated in other sub-routines using an appropriate choice of kinetic modeling. The beam is then vacuum propagated through the remaining $\Delta L/2$. This procedure is repeated until the beam reaches the end of the cavity.

c. Fortran

Argument List

NCAV = Cavity identity number (1, 2, 3, ... N)

ILR = identifies the direction of propagation through the cavity:

-1 => right to left

+1 => left to right

NEWCAV = A parameter that identifies whether the cavity has been entered before.

INIT = .True. if it is the first interaction of a given run
= .False. if it is the second or subsequent interaction.

NSTE = Controlling parameter for subroutine STEP. If the geometric beam is converging or diverging, variable area mesh propagation (VAMP) should be used.

NSTE = 1 Constant mesh with setup

= 2 VAMP with setup (exit at end)

= 3 VAMP (setup and remain in VAMP)

= 4 VAMP (uses existing setup and exits)

= 5 VAMP (uses existing setup and does not exit)

IN = Input data set number or file from which data is to be read

RESTRT = .True. if initial beam is read in from unit IB

= .False. if analytical initial field is desired

NPLT = Controls plotting within cavity:

= 0 No plot

= 1 Print field before and after gain and gain coefficient

ZLI = Incoming propagation distance to cavity endwall

(Additional vacuum propagation distance)

ZLO = Exit propagation distance to cavity endwall

(Additional vacuum propagation distance)

Note: None of the parameters in the argument list is redefined by subroutine CAVITY.

Common variables altered:

- US - the intensity array
- PPD - interpolated power density
- CDUM - interpolated gain/phase transmission element
- XCAV - cavity coordinate array
- GFACT - define by namelist CAVTY2
- CFIL - redefined by its equivalence with Power Density array
- CU - the complex field - modified by propagation and the application of the cavity transmission function
- CG - defined for the first pass, read in for subsequent passes (Cavity gain/phase (G/P) array at each station within the cavity)

Namelist/CAVTY 2

CAVTY2 is used to initialize the cavity physical properties. The namelist is as follows:

```

NAMLIST/ CAVTY2 /XLEN,YLEN,ZLEN,XMCAV,YMCAV,NODX,NODY,NOSEG,
* FLAG,MREST,NGTYPE,NOPL0T,IUSE,IPUEN,T1,T2,T3,TN2,TS,PS,V,
* PARCH,XN2,XC02,XH20,XC0,X02,TITLE,ALFA,ACP,VELTY,TTEMP,ANGL,
* AVGAIN,GFACT
C
C XLEN IS LENGTH OF CAVITY IN FLOW DIRECTION
C YLEN IS LENGTH OF CAVITY ACROSS NOZZLES
C ZLEN IS LENGTH OF CAVITY IN OPTICAL DIRECTION
C XMCAV IS THE X-DIST OF OPTICAL AXIS FROM NOZZLE EXIT PLANE
C YMCAV IS THE Y-DIST OF OPTICAL AXIS TO CAVITY AXIS
C NODX IS NUMBER OF GRID POINTS ALONG XLEN
C NODY IS NUMBER OF GRID POINTS ALONG YLEN
C NOSEG IS NUMBER OF SEGMENTS, MAXIMUM OF 5 PER CAVITY
C FLAG IS PARAMETER WHICH CONTROLS SELECTION OF DENSITY FIELD
C = 1. SP=1,CUNTOURED SIDEWALL
C = 2. SP=1, FLAT SIDEWALL
C = 3. ALL DENSITY
C = 4. MOD=6, XLS=1
C = 5. INPUT FROM CARDS ON DATA SET...IN...
C = 6. SAME SPLINE CO-EFF THAT WERE READ IN FIVE
C =8. RUN 112 AT T=1.6 SEC RIGHT STAGE BOTH WALLS
C =8.1 READ NAMFLIST DENS8 FOR RIGHT STAGE
C =9. RUN 109 AT T=1.8 SEC LEFT STAGE BOTH WALLS
C =9.1 READ NAMFLIST DENS9 FOR LEFT STAGE
C =10. READ DENSITY FIELD FROM UNIT 30
C =11. READ DENSITY FIELD FROM UNIT 31
C MREST IS A FLAG FOR COMPUTING A RESTRAINED GAIN...IF
C = 1 READ OFF THE BIG G BUT USE NEW DENSITY FIELD
C = 0 THEN TAKE THE CO-EFF AS THEY NOW EXIST
C
C NGTYPE = 2...THERMAL BLOOMING FOR MULTI-BEAM
C = 1...FULL BLOWN KINETICS...GOL
C = 0 SIMPLF CLOSED FORM E.A.S. GOL KINETICS
```



```

C
C   NGPLOT = 0  NO PLOTS OF GAIN INSIDE THE CAVITY
C           = 1  PLOT A SLICE THROUGH THE CAVITY
C           = 2  ISO-AMPLITUDE OF GAIN IS PLOTTED
C           = 3  GET BOTH PLOTS
C           =-1  GET ALL POSSIBLE PLOTS
C
C   IPDEN = 0  NO PLOT OF POWER DENSITY AT EACH SLICE
C           = 1  SLICE PLOT OF PWR DENS
C           = 2  ISO- INTENSITY PLOT FOR CAVITY
C           = 3  ALL FOR THE MONEY
C   IUSE  = -1  NO FUSE NO PLOTS NO NOTIN
C           = 0  NO FUSE ANALYSIS. BUT DENSITY GOULY PLOTS (AERO)
C           = 1  FUS ANALYSIS...NO PLOTS
C           = 2  FUS IS USED (RHYME?) AS IS ISO-PLOTS
C           = 3  FUS. ISO-PLOTS OF FUS AND RESULTANT FUSE AND AERO
C   TITLE IS THE TITLE TO APPEAR ON THE CAVITY GOULIES & GOULESESS
C
C   T1 IS VIBRATIONAL TEMPERATURE OF OUV      AT NEP, DEG K
C   T2 IS VIBRATIONAL TEMPERATURE OF UVO      AT NEP, DEG K
C   T3 IS VIBRATIONAL TEMPERATURE OF VOO      AT NEP, DEG K
C   TN2 IS VIBRATIONAL TEMPERATURE OF NITROGEN AT NEP, DEG K
C   TS IS STATIC TEMPERATURE IN CAVITY AT NEP, DEG K
C   PS IS STATIC PRESSURE IN CAVITY AT NEP, ATM.
C   V    IS FLOW VELOCITY IN CAVITY AT NEP, CM/SEC
C   PRCH IS P-BRANCH TRANSITION
C   AN2 IS MOLE FRACTION OF NITROGEN
C   XC02 IS MOLE FRACTION OF CARBON DIOXIDE
C   XH2O IS MOLE FRACTION OF WATER
C   XC0  IS MOLE FRACTION OF CARBON MONOXIDE
C   XO2  IS MOLE FRACTION OF OXYGEN
C ***** THERMAL BLOOMING MULTI-BEAM CAVITY *****
C   ALFA IS THE MEDIUM ABSORB CO-EFF IN CM-1
C   ACP IS THE MEDIUM SPECIFIC HEAT IN J/GM-DEG K
C   TTEMP IS THE MEDIUM TEMPERATURE IN DEG K
C   VELTY IS THE VELOCITY OF MEDIUM...IF .LT. 1. THEN THE FREE
C   CONVECTION VELOCITY IS CALCULATED AND USED
C   ANGLE IS THE ANGLE OF FLOW RELATIVE TO N.E.P. 0. IS LIKE CAVITY
C   (IF. AWAY FROM N.E.P.) AND 180. IS THE OTHER DIRECTION
C *****
C
C   AVGAIN IS THE AVERAGE OF GAIN CO-EFFICIENTS...HOPE FAST CONVERGE
C
C

```

SUBROUTINE CAVITY 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE CAVITY(INCAVN,ILR,NEWCAV,INIT,NSTE,IN,NESTRT,NPLT, CAVITY 2
A ZLI,ZLJ) CAVITY 3
C   GUL CAVITY MODEL CAVITY 4
C   THIS ROUTINE APPLIES THE EFFECTS OF A GUL CAVITY TO THE COMPLEX CAVITY 5
C   FIELD CAVITY 6
C   LEVEL 2: CU*AC,XCAV,PDD,PPD,CUR,CB,US CAVITY 7
C   LEVEL 2:PU COMH2 8
C   COMMON /CUS/ US(1700) COMH1 9
C   COMMON /CAVX/ PDD,XCAV,CUM COMH1 10
C   COMMON /MHPHOP/MADCUR,ANGA,ANGY CAVITY 11
C   COMMON /GFACT1/ GFACT(2) LMOPI 12
C   COMMON /RAY/NUM,NREG,MAPTH CAVITY 13
C   COMMON /CAV2/ AC(5),YC(5),ZC(5),NA(5),NY(5),NS(5),XMC(5),YMC(5), CAVITY 14

```

2	NGTYM9(5), NGPLU9(5), IUSY(5), IPUE9(5),	CAVITY	12
3	SSGAIN(190,5),SATIN(5),DELTA(5),NMUS(5),	CAVITY	13
4	VEL(5),GAM(5),XMACH(5),TV1(5),TV2(5),TV3(5),TVN2(5),TSCAV(5),	CAVITY	14
5	MSCAV(5),PH(5),FN2(5),FLU2(5),FM20(5),FCU(5),FU2(5),TITLE(20),	CAVITY	15
6	AVG(5), NSYM	CAVITY	16
	COMMON/MELT/CU(16384),CFIL(16512),X(128),XL,NPTS,NPY,DX,UMY	CAVITY	17
	COMMON /CG/ CG(17100)	CIUDENS	1
	DIMENSION TPASS(290), TMU(280), PD(17100), PMU(17100),	CIUDENS	2
X	PMU(2), ACAV(190), CUM(32768)	CIUDENS	3
	COMPLEX CU,CFIL,CG,CAKAY,CDUM	CAVITY	20
	LOGICAL INIT,MESTHT	CAVITY	21
	EQUIVALENCE (CUM(1),CU(1))	CAVITY	22
	EQUIVALENCE (PPD(1),CU(1)), (CFIL(1),PD(1))	CIUDENS	4
	DATA GFACR / 1. /	LHUPI	10
	DATA XLEN,YLEN,ZLEN,XMCAV,YMCAV,NOUX,NOUY,NOSEG.	CAVITY	25
X	FLAG,MHST,NGTYPE,NGPLUT,IUSE,IPDEN,T1,T2,T3,TN2,TS,MS,V.	CAVITY	26
X	PBRCH,XN2,ACU2,XM2U,XCUM,ACU2,ALFA,ACP,VELTY,ITEMP,ANGL,AVGAIN	CAVITY	27
X	/3*0.0,2*0.3,0.0,3*0.-1.0,19*0.0/	CAVITY	28
C		CAVITY	29
	NAMelist/ CAVITYZ /XLEN,YLEN,ZLEN,XMCAV,YMCAV,NOUX,NOUY,NOSEG.	CAVITY	30
X	FLAG,MHST,NGTYPE,NGPLUT,IUSE,IPDEN,T1,T2,T3,TN2,TS,MS,V.	CAVITY	31
X	PBRCH,XN2,ACU2,XM2U,XCUM,ACU2,TITLE,ALFA,ACP,VELTY,ITEMP,ANGL,	CAVITY	32
X	AVGAIN, GFACR	LHUPI	11
C		CAVITY	34
C	XLEN IS LENGTH OF CAVITY IN FLOW DIRECTION	CAVITY	35
C	YLEN IS LENGTH OF CAVITY ACROSS NOZZLES	CAVITY	36
C	ZLEN IS LENGTH OF CAVITY IN OPTICAL DIRECTION	CAVITY	37
C	XMCAV IS THE X-DIST OF OPTICAL AXIS FROM NOZZLE EXIT PLANE	CAVITY	38
C	YMCAV IS THE Y-DIST OF OPTICAL AXIS TO CAVITY AXIS	CAVITY	39
C	NOUX IS NUMBER OF GRID POINTS ALONG XLEN	CAVITY	40
C	NOUY IS NUMBER OF GRID POINTS ALONG YLEN	CAVITY	41
C	NOSEG IS NUMBER OF SEGMENTS, MAXIMUM OF 3 PER CAVITY	CAVITY	42
C	FLAG IS PARAMETER WHICH CONTROLS SELECTION OF DENSITY FIELD	CAVITY	43
C	= 1. SH=1, CUNTOUMED SIDEWALL	CAVITY	44
C	= 2. SH=1, FLAT SIDEWALL	CAVITY	45
C	= 3. ALL DENSITY	CAVITY	46
C	= 4. MOU=0, XLS=1	CAVITY	47
C	= 5. INPUT FROM CARDS ON DATA SET...IN...	CAVITY	48
C	= 6. SAME SPLINE CO-EFF THAT WERE READ IN FIVE	CAVITY	49
C	=8.1 HEAD NUM 112 AT T=1.8 SEC RIGHT STAGE BOTH WALLS	SOU77CY1	3
C	=8.1 HEAD NAMELIST DENS8 FOR RIGHT STAGE	SOU77CY1	4
C	=9.1 HEAD NUM 109 AT T=1.8 SEC LEFT STAGE BOTH WALLS	SOU77CY1	5
C	=9.1 HEAD NAMELIST DENS9 FOR LEFT STAGE	SOU77CY1	6
C	=10. HEAD DENSITY FIELD FROM UNIT 30	SOU77CY1	7
C	=11. HEAD DENSITY FIELD FROM UNIT 31	SOU77CY1	8
C	MHST IS A FLAG FOR COMPUTING A MODERATED GAIN...IF	CAVITY	50
C	= 1 HEAD OFF THE SIG 0 BUT USE NEW DENSITY FIELD	CAVITY	51
C	= 0 THEN TAKE THE CO-EFF AS THEY NUM EXIST	CAVITY	52
C		CAVITY	53
C	NGTYPE = 2...THERMAL HLOOMING FOR MULTI-BEAM	CAVITY	54
C	= 1...FULL BLOWN KINETICS...GOL	CAVITY	55
C	= 0 SIMPLE CLOSED FORM E.A.S. GOL KINETICS	CAVITY	56
C		CAVITY	57
C	NGPLUT = 0 NO PLOTS OF GAIN INSIDE THE CAVITY	CAVITY	58
C	= 1 PLOT A SLICE THROUGH THE CAVITY	CAVITY	59
C	= 2 ISO-AMPLITUDE OF GAIN IS PLOTTED	CAVITY	60
C	= 3 GET BOTH PLOTS	CAVITY	61
C	=-1 GET ALL POSSIBLE PLOTS	CAVITY	62
C		CAVITY	63
C	IPDEN = 0 NO PLOT OF POWER DENSITY AT EACH SLICE	CAVITY	64
C	= 1 SLICE PLOT OF POW DENS	CAVITY	65
C	= 2 ISO- INTENSITY PLOT FOR CAVITY	CAVITY	66
C	= 3 ALL FOR THE MONEY	CAVITY	67
C	IUSE = -1 NO FUSE NO PLOTS NO NOTHING	CAVITY	68
C	= 0 NO FUSE ANALYSIS, BUT DENSITY GOULY PLOTS (AENU)	CAVITY	69
C	= 1 FUMS ANALYSIS...NO PLOTS	CAVITY	70
C	= 2 FUMS IS USED (WHYME?) AS IS ISO-PLOTS	CAVITY	71
C	= 3 FUMS, ISO-PLOTS OF FUMS AND RESULTANT FUSE AND AENU	CAVITY	72
C	TITLE IS THE TITLE TO APPEAR ON THE CAVITY GOULIES & GOULESS	CAVITY	73
C		CAVITY	74
C	T1 IS VIBRATIONAL TEMPERATURE OF UOV AT NEP, DEG R	CAVITY	75
C	T2 IS VIBRATIONAL TEMPERATURE OF UOV AT NEP, DEG R	CAVITY	76

```

C      T3 IS VIBRATIONAL TEMPERATURE OF V00          AT NEP, DEG K          CAVITY 77
C      TN2 IS VIBRATIONAL TEMPERATURE OF NITROGEN    AT NEP, DEG K          CAVITY 78
C      TS IS STATIC TEMPERATURE IN CAVITY AT NEP, DEG K          CAVITY 79
C      PS IS STATIC PRESSURE IN CAVITY AT NEP, ATM.          CAVITY 80
C      V      IS FLOW VELOCITY IN CAVITY AT NEP, CM/SEC          CAVITY 81
C      PBNCH IS P-BRANCH TRANSITION          CAVITY 82
C      AN2 IS MOLE FRACTION OF NITROGEN          CAVITY 83
C      ACU2 IS MOLE FRACTION OF CARBON DIOXIDE          CAVITY 84
C      AM2O IS MOLE FRACTION OF WATER          CAVITY 85
C      ACU IS MOLE FRACTION OF CARBON MONOXIDE          CAVITY 86
C      XU2 IS MOLE FRACTION OF OXYGEN          CAVITY 87
C      ***** THERMAL BLOOMING MULTI-BEAM CAVITY *****          CAVITY 88
C      ALFA IS THE MEDIUM ABSORB CO-EFF IN CM-1          CAVITY 89
C      ACP IS THE MEDIUM SPECIFIC HEAT IN J/GM-DEG K          CAVITY 90
C      TTEMP IS THE MEDIUM TEMPERATURE IN DEG K          CAVITY 91
C      VELTY IS THE VELOCITY OF MEDIUM...IF .LT. 1, THEN THE FREE          CAVITY 92
C      CONVECTION VELOCITY IS CALCULATED AND USED          CAVITY 93
C      ANGL IS THE ANGLE OF FLOW RELATIVE TO N.E.P. 0. IS LIKE CAVITY          CAVITY 94
C      (IE. AWAY FROM N.E.P.) AND 180. IS THE OTHER DIRECTION          CAVITY 95
C      *****          CAVITY 96
C      AVGAIN IS THE AVERAGE OF GAIN CO-EFFICIENTS...HOPE FAST CONVERGE          CAVITY 97
C      CAVITY 98
C      CAVITY 99
C      CAVITY 100
C      CAVITY 101
C      *** TEST TO SEE IF BEEN IN THIS CAVITY BEFORE          CAVITY 102
C      IF (.NOT. INIT.OM.NE.CAV.EQ.0) GO TO 50          CAVITY 103
C      PI = 3.141592          CAVITY 104
C      NSYM = 0          CAVITY 105
C      IF (.NOT. NPTS) NSYM=1          CAVITY 106
C      NNSYM=NSYM          CAVITY 107
C      NUB = NPTS*NPY          CAVITY 108
C      MNEST = 0          CAVITY 109
C      HEAD(IN.CAVITY2)          CAVITY 110
C      READ (IN+1203) TITLE          CAVITY 111
C      1203 FORMAT (20A)          CAVITY 112
C      WRITE(6,200)          CAVITY 113
C      200 FORMAT(3YHU          CAVITY 114
C      X          3YHU          CAVITY 115
C      X          3YHU          CAVITY 116
C      WRITE(6,100) TITLE,YLEN,ZLEN,NNUX,NNUY,NUSEG          CAVITY 117
C      100 FORMAT(18H0CAVITY GEOMETRY FOM ,20A//1X,7MXLEN = ,G12.5,4X,7MYLEN          CAVITY 118
C      X = ,G12.5,4X,7MZLEN = ,G12.5,4X,6MNNUX = ,16,4X,7MNNUY = ,15,4X,          CAVITY 119
C      8MNNUSEG = ,12)          CAVITY 120
C      WRITE(6,101) MCAV,VMCAV          CAVITY 121
C      101 FORMAT(25H0LOCATION OF OPTICAL AXIS//1X,8MAMCAV = ,G12.5,4X,          CAVITY 122
C      X 8MYMCAV = ,G12.5)          CAVITY 123
C      IF (.NOT. NGTYPE.EQ.2) GO TO 106          CAVITY 124
C      WRITE(6,102) TS,PS,V,PBNCH          CAVITY 125
C      102 FORMAT(18H0CAVITY CONDITIONS//1X,5MTS = ,G12.5,4X,5MPS = ,G12.5,          CAVITY 126
C      4X,11HVELOCITY = ,G12.5,4X,9MP-PBRANCH ,F3.0)          CAVITY 127
C      WRITE(6,103) AN2,ACU2,AM2O,ACU,XU2          CAVITY 128
C      103 FORMAT(12H0COMPOSITION//1X,6MAN2 = ,G12.5,4X,7MACU2 = ,G12.5,4X,          CAVITY 129
C      7MAM2O = ,G12.5,4X,6MACU = ,G12.5,4X,6MXU2 = ,G12.5)          CAVITY 130
C      LOAD CAVITY PARAMETERS INTO APPROPRIATE STORAGE ARRAYS          CAVITY 131
C      TV1(NCAVN)=T1          CAVITY 132
C      TV2(NCAVN)=T2          CAVITY 133
C      TV3(NCAVN)=T3          CAVITY 134
C      TVN2(NCAVN)=TN2          CAVITY 135
C      TSCAV(NCAVN)=TS          CAVITY 136
C      WRITE(6,104) TN2,T1,T2,T3          CAVITY 137
C      104 FORMAT(25H0VIBRATIONAL TEMPERATURES//1X,6MTN2 = ,G12.5,4X,5MT1 = ,          CAVITY 138
C      XG12.5,4X,5MT2 = ,G12.5,4X,5MT3 = ,G12.5)          CAVITY 139
C      GO TO 107          CAVITY 140
C      106 MNEST = 2          CAVITY 141
C      TV1(NCAVN)=ALFA          CAVITY 142
C      TV2(NCAVN)=ACP          CAVITY 143
C      TV3(NCAVN)=VELTY          CAVITY 144
C      TVN2(NCAVN)=TTEMP          CAVITY 145
C      TSCAV(NCAVN)=ANGL          CAVITY 146
C      WRITE(6,133) ALFA,ACP,VELTY,TTEMP,ANGL

```

```

193 FORMAT (//6TH THERMAL BLOOMING ANALYSIS OF MULTI-BEAM SYSTEM...C CAVITY 147
CONSTANTS ARE ://7M ALFA =.0125*GM CP =.FB.4*17M FLOW VELOCITY = CAVITY 148
X,FB.4, UM TEMP = .FB.4, LUM ANGLE = .FB.4 // ) CAVITY 149
107 YC( NCAVN ) = YLEN CAVITY 150
XC( NCAVN ) = XLEN CAVITY 151
ZC( NCAVN ) = ZLEN CAVITY 152
AVG( NCAVN ) = AVGAIN CAVITY 153
XMC( NCAVN ) = XMCAV CAVITY 154
YMC( NCAVN ) = YMCAV CAVITY 155
NA( NCAVN ) = NOXA CAVITY 156
NY( NCAVN ) = NOYA CAVITY 157
NS( NCAVN ) = NOSEG CAVITY 158
GFACT( NCAVN ) = GFACIM LHUPI 12
DCZ = ZC( NCAVN ) / NS( NCAVN ) CAVITY 159
UCA = XC( NCAVN ) / NA( NCAVN ) CAVITY 160
UCY = YC( NCAVN ) / NY( NCAVN ) CAVITY 161
NSA = NS( NCAVN ) CAVITY 162
NTA = NY( NCAVN ) / ( NSYM + 1 ) CAVITY 163
NAA = NA( NCAVN ) CAVITY 164
MUT = NAA * NYA CAVITY 165
NGTYP9( NCAVN ) = NGTYPE CAVITY 166
NGPLU9( NCAVN ) = NGPLUT CAVITY 167
IUS9( NCAVN ) = IUSE CAVITY 168
IPUE9( NCAVN ) = IPUEN CAVITY 169
MSCAV( NCAVN ) = PS CAVITY 170
VEL( NCAVN ) = V CAVITY 171
PB( NCAVN ) = PBHCH CAVITY 172
CARAY = CMPLX( U, .2 * PI / WL ) CAVITY 173
TUPI = L = 2. * PI / WL SUQ77CY1 9
FN2( NCAVN ) = AN2 CAVITY 174
FCO2( NCAVN ) = XCO2 CAVITY 175
FM20( NCAVN ) = XM20 CAVITY 176
FCO( NCAVN ) = XCO CAVITY 177
FO2( NCAVN ) = AU2 CAVITY 178
IBASE = 10 * ( NCAVN - 1 ) * 11 CAVITY 179
IF ( NGTYPE .EQ. 2 ) GO TO 108 CAVITY 180
C CALCULATE SMALL SIGNAL GAIN AS A FUNCTION OF X CAVITY 181
CALL GAINXY( PD, US, NCAVN, 1 ) CAVITY 182
WRITE ( 7 ) ( CG( IZ ), IZ = 1, NUB ) CAVITY 183
HE = INU 7 CAVITY 184
MMU = MMUS( NCAVN ) CAVITY 185
C CALCULATE CAVITY DENSITY FIELD AS A FUNCTION OF X AND Y CAVITY 186
CALL DENS( FLAG, MMU, XLEN, YLEN, UCZ, NAA, NYA, 1, IN, NNSYM ) CAVITY 187
C STORE DENSITY FIELD ON DIRECT ACCESS FILE CAVITY 188
WRITE( IBASE ) ( MPD( IZ ), IZ = 1, MUT ) CAVITY 189
HE = INU IBASE CAVITY 190
C CAVITY 191
C IF RESTARTING FROM A PREVIOUS RUN, THEN SKIP THE INITIAL CAVITY 192
C GUESS AT GAIN CAVITY 193
C CAVITY 194
108 IF ( NESTRT .AND. MNEST .NE. 1 ) GO TO 49 CAVITY 195
DO 10 NNS = 1, NSA CAVITY 196
XCLO = -DCX / 2. CAVITY 197
IBASE = IBASE + 1 CAVITY 198
IF ( MNEST .NE. 1 ) GO TO 20 CAVITY 199
READ( IBASE ) ( CG( IZ ), IZ = 1, MUT ) CAVITY 200
HE = INU IBASE CAVITY 201
C GENERATE COMPLEX GAIN ARRAYS CAVITY 202
20 XMUL1 = UCZ / 6. CAVITY 203
DO 11 IX = 1, NAA CAVITY 204
XCLO = DCX * XCLO CAVITY 205
GUP = SSUAIN( IX, NCAVN ) CAVITY 206
AMULTN = EXP( XMUL1 * GUP ) CAVITY 207
DO 11 IY = 1, NYA CAVITY 208
IZ = IX * ( IY - 1 ) * NAA CAVITY 209
MMIM = TUPI * WL * MPD( IZ ) SUQ77CY1 10
IF ( MNEST .EQ. 0 ) CAVITY 210
ACG( IZ ) = XMULTN * CMPLX( COS( MMIM ), SIN( MMIM ) ) SUQ77CY1 11
C X CG( IZ ) = EXP( GUP * UCZ / 6. ) * CEAP( CARAY * MPD( IZ ) ) CAVITY 212
IF ( MNEST .EQ. 1 ) CAVITY 213
X CG( IZ ) = CABS( CG( IZ ) ) * CMPLX( COS( MMIM ), SIN( MMIM ) ) SUQ77CY1 12
C X CG( IZ ) = CABS( CG( IZ ) ) * CEAP( CARAY * MPD( IZ ) ) CAVITY 215

```

IF (MHST.EQ.2)	CAVITY	216
A CU(IZ) = CMPLX(1.0,0)	CAVITY	217
11 CONTINUE	CAVITY	218
WRITE(1BASE) (CU(IZ),IZ=1,MUT)	CAVITY	219
10 NEW(ND) 1BASE	CAVITY	220
49 READ (7) (CU(IZ),IZ=1,NUM)	CAVITY	221
NEW(ND) 7	CAVITY	222
C APPLICATION OF CAVITY TRANSMISSION FUNCTIONS TO COMPLEX FIELD	CAVITY	223
50 IAS=NS(NCAVN)	CAVITY	224
NYA=NY(NCAVN)/(NSYM+1)	CAVITY	225
NXA=NA(NCAVN)	CAVITY	226
MUT = NXA*NYA	CAVITY	227
C *** FIRST TIME THROUGH THIS CAVITY, ZERO AVERAGE INTENSITY ARRAY	CAVITY	228
IF (NEWCAV.EQ.0) GO TO 51	CAVITY	229
C CALL ZERO(PU(1),PU(MUT))	CAVITY	230
DO 485 IZERU=1,MUT	CAVITY	231
485 PU(IZERU)=0.	CAVITY	232
1BASE=10*(NCAVN-1)+1+5	CAVITY	233
NCULD = 0	CAVITY	234
DO 53 IZ=1,NSA	CAVITY	235
1BAS=1BASE+IZ	CAVITY	236
WRITE (1BAS) (PU(IZ),IZ=1,MUT)	CAVITY	237
53 NEW(ND) 1BAS	CAVITY	238
51 1BASE = 10*(NCAVN-1)+1	CAVITY	239
IF (NCAVN.EQ.0) GO TO 26	CAVITY	240
UX = AC(NCAVN)/NXA	CAVITY	241
DY = YC(NCAVN)/NY(NCAVN)	CAVITY	242
C ESTABLISH CAVITY INTERPOLATION ARRAY (TPASS)	CAVITY	243
TPASS(1) = UX	CAVITY	244
TPASS(2) = DY	CAVITY	245
TPASS(3) = NYA*.001	CAVITY	246
TPASS(4) = NXA*.001	CAVITY	247
TPASS(5) = (DY-YC(NCAVN))/2. + YMC(NCAVN)	CAVITY	248
TPASS(5+NYA) = UX/2. - XMC(NCAVN)	CAVITY	249
DO 5 I = 2,NYA	CAVITY	250
5 TPASS(4+I) = TPASS(3+I) + DY	CAVITY	251
DO 6 N = 2,NXA	CAVITY	252
6 TPASS(4+NYA+N) = TPASS(3+NYA+N) + UX	CAVITY	253
NCULD = NCAVN	CAVITY	254
26 NST=NSTE	CAVITY	255
IOUT=1	CAVITY	256
DCZ = ZC(NCAVN)/NSA	CAVITY	257
C PROPAGATE TO FIRST GAIN/PHASE SEGMENT	CAVITY	258
IF (NSTE.EQ.3.OR.NSTE.EQ.5) IOUT=0	CAVITY	259
IF (NSTE.EQ.3) NST=2	CAVITY	260
C IF (NSTE.GE.4.AND.(DCZ/2.+ZLI).GT.1.0) CALL COME(DCZ/2.0+ZLI,0.0)	CAVITY	261
IF (NSTE.GE.4.AND.(DCZ/2.+ZLI).GT.1.0) CALL STEP(DCZ/2.0+ZLI,	CAVITY	262
A HADCUM+.1+.1,NST,0.0,ANGX,ANGY,0.1)	CAVITY	263
IF (NSTE.LE.3.AND.(DCZ/2.+ZLI).GT.1.0)	CAVITY	264
1CALL STEP(DCZ/2.0+ZLI,HADCUM+.1+.1,NST, 0.0,ANGX,ANGY,0.0)	CAVITY	265
MEMORY=0	CAVITY	266
IF (NSTE.LE.3.AND.(DCZ/2.+ZLI).LE.1.0)MEMORY=1	CAVITY	267
DO 55 JNS=1,NSA	CAVITY	268
IB = 0	CAVITY	269
IF (ILH.LI.0) IB=NS(NCAVN)+1	CAVITY	270
IAOU = JNS*ILH+1	CAVITY	271
XFACT=1.	CAVITY	272
IF (INLEG.NE.0) XFACT=1./WNUM**2	CAVITY	273
IUPD = IAU+5+1BASE	CAVITY	274
C ESTABLISH FIELD INTERPOLATION ARRAY	CAVITY	275
TPU(1) = A(2)-X(1)	CAVITY	276
TPU(2) = TPU(1)	CAVITY	277
TPU(3) = NPY	CAVITY	278
TPU(4) = NPTS	CAVITY	279
DO 54 IPJ=1,NPY	CAVITY	280
54 TPU(IPJ+4) = X(IPJ)*UMY	CAVITY	281
DO 82 IPJ=1,NPTS	CAVITY	282
82 TPU(IPJ+NPY+4) = A(IPJ)*UMA	CAVITY	283
C *** COMPUTE INTENSITY INCIDENT UPON SEGMENT	CAVITY	284
DO 61 MA=1,NUM	CAVITY	285

01	US(MX) = (CUM(2*MX-1)**2 + CUM(2*MX)**2) * XFACT	CAVITY	286
	WRITE (7) (US(IZ),IZ=1,NUB)	CAVITY	287
	HE=INU 7	CAVITY	288
	IUCG = IADD*IBASE	CAVITY	289
	HEAD(IUCG) (CG(IZ),IZ=1,NU)	CAVITY	290
	HE=INU IUCG	CAVITY	291
	IF (INPLT.EQ.0) GO TO 68	CAVITY	292
C	PLUT FIELD INCIDENT ON GAIN/PHASE SEGMENT	CAVITY	293
	WRITE (6,69) NCAVN,IAUD	CAVITY	294
69	FORMAT(J6M1) ***** E-M FIELD IN CAVITY NUMBER ,IZ,19M AT 5	CAVITY	295
	ARGUMENT # ,IZ,41M BEFORE GAIN HAS BEEN APPLIED ***** /)	CAVITY	296
	K=1	CAVITY	297
	UMAX=0.0	CAVITY	298
	CALL OUTPUT(CU,NPY,NPTS,X,K,UMAX,.TRUE...FALSE...FALSE.)	CAVITY	299
C	PLUT GAIN PROFILE THROUGH CENTER OF CAVITY	CAVITY	300
	WRITE (6,67) NCAVN,IAUD	CAVITY	301
67	FORMAT(90M1) CG(I,J) PLOTTED IN THE X-DIRECTION THROUGH THE CENTER	CAVITY	302
	X OF THE CAVITY, FOR CAVITY # ,IZ,19M SEGMENT # ,IZ)	CAVITY	303
	DELXC=X(NCAVN)/NX(NCAVN)	CAVITY	304
	XCAV(1)=UDELXC/2.	CAVITY	305
	UU 667 KCX=2,NAA	CAVITY	306
667	XCAV(KCX)=XCAV(KCX-1)*UDELXC	CAVITY	307
	K=1	CAVITY	308
	UMAX=0.0	CAVITY	309
	CALL OUTPUT(CG,NY(NCAVN),NX(NCAVN),XCAV,K,UMAX,.TRUE...FALSE..	CAVITY	310
	X .FALSE.)	CAVITY	311
68	IZ=0	CAVITY	312
C	APPLY CAVITY TRANSMISSION TO COMPLEX FIELD	CAVITY	313
	DU 58 JY=1,NPY	CAVITY	314
	DU 58 JX=1,NPTS	CAVITY	315
	CALL INTERP(TPASS,X(JX)*UMA,X(JY)*UMY,CG,2,CDUM,NNSYM)	COMRI	48
	IZ = IZ+1	CAVITY	317
58	CU(IZ) = CDUM*CU(IZ)	CAVITY	318
	HEAD (7) (US(IZ),IZ=1,NUB)	CAVITY	319
	HE=INU 7	CAVITY	320
C	CALCULATE SUM OF INTENSITIES BEFORE AND AFTER GAIN/PHASE SEGMENT	CAVITY	321
	DU 64 JY=1,NUB	CAVITY	322
64	US(JY) =(CUM(2*JY-1)**2 + CUM(2*JY)**2) * XFACT*US(JY)	CAVITY	323
	HEAD (IDPU) (PU(IZ),IZ=1,MUT)	CAVITY	324
	HE=INU IUPU	CAVITY	325
	IF (INPLT.EQ.0) GO TO 73	CAVITY	326
C	PLUT FIELD LEAVING GAIN/PHASE SEGMENT	CAVITY	327
	WRITE (6,39) NCAVN,IAUD	CAVITY	328
39	FORMAT(///J6M1) ***** E-M FIELD IN CAVITY NUMBER ,IZ,19M AT 5	CAVITY	329
	ARGUMENT # ,IZ,40M AFTER GAIN HAS BEEN APPLIED ***** /)	CAVITY	330
	K=1	CAVITY	331
	UMAX=0.0	CAVITY	332
	CALL OUTPUT(CU,NPY,NPTS,X,K,UMAX,.TRUE...FALSE...FALSE.)	CAVITY	333
73	TPUMIN=TPD(5)	CAVITY	334
C	INTERPOLATE POWER DENSITIES UNTO CAVITY GRID, SUM WITH RESULTS	CAVITY	335
C	OF PREVIOUS PASSES AND STORE	CAVITY	336
	DU 57 INY=1,NYA	CAVITY	337
	TTESY=TPASS(+INX)	CAVITY	338
	IF (TTESY.LI.TPUMIN) GO TO 57	CAVITY	339
	DU 56 INX=1,NXA	CAVITY	340
	TTESX=TPASS(+NYA+INX)	CAVITY	341
	CALL INTERP(TPD,TTESX,TTESY,US,1,POD,NNSYM)	COMRI	49
	IZ = INX+(INX-1)*NXA	CAVITY	343
56	PU(IZ) = PU(IZ)+POD(1)/2.	CAVITY	344
57	CONTINUE	CAVITY	345
	WRITE (IDPD) (PD(IZ),IZ=1,MUT)	CAVITY	346
	HE=INU IUPD	CAVITY	347
C	PROPAGATE TO NEXT GAIN/PHASE SEGMENT	CAVITY	348
C	IF (JNS.NE.NSA.AND.MEMORY.EQ.0)CALL CUME(ICZ,0,0)	CAVITY	349
	IF (JNS.NE.NSA.AND.MEMORY.EQ.0)CALL STEP(ICZ,RAUCUR,.1,.1,NSI,0,	CAVITY	350
	X 0,ANGX,ANGY,0,1)	CAVITY	351
	IF (JNS.NE.NSA.AND.MEMORY.EQ.1)	CAVITY	352
	ICALL STEP(ICZ,RAUCUR,.1,.1,NSI, 0,0,ANGX,ANGY,0,0)	CAVITY	353
	MEMORY=0	CAVITY	354
C	PROPAGATE OUT OF CAVITY	CAVITY	355
C	IF (JNS.EQ.NSA.AND.(ICZ/2.*ZLU).GT.1.0)CALL CUME(ICZ/2.0*ZLU,1OUT,0	CAVITY	356

```

C   X)
    IF (JNS.EU.NSA.AND.(DCZ/2.*ZLO).GT.1.0) CALL STEP(DCZ/2.0*ZLO,
X   RADCUR=.1,.1,NSI,IOUT,0,ANGA,ANGY,0,1)
55 CONTINUE
    RETURN
    END

```

```

CAVITY 357
CAVITY 358
CAVITY 359
CAVITY 360
CAVITY 361
CAVITY 362

```

8. SUBROUTINE CENBAR

a. Purpose -- This subroutine is used by QUAL to find the centroid coordinates of the far-field beam. Figure 20 describes subroutine CENBAR organization.

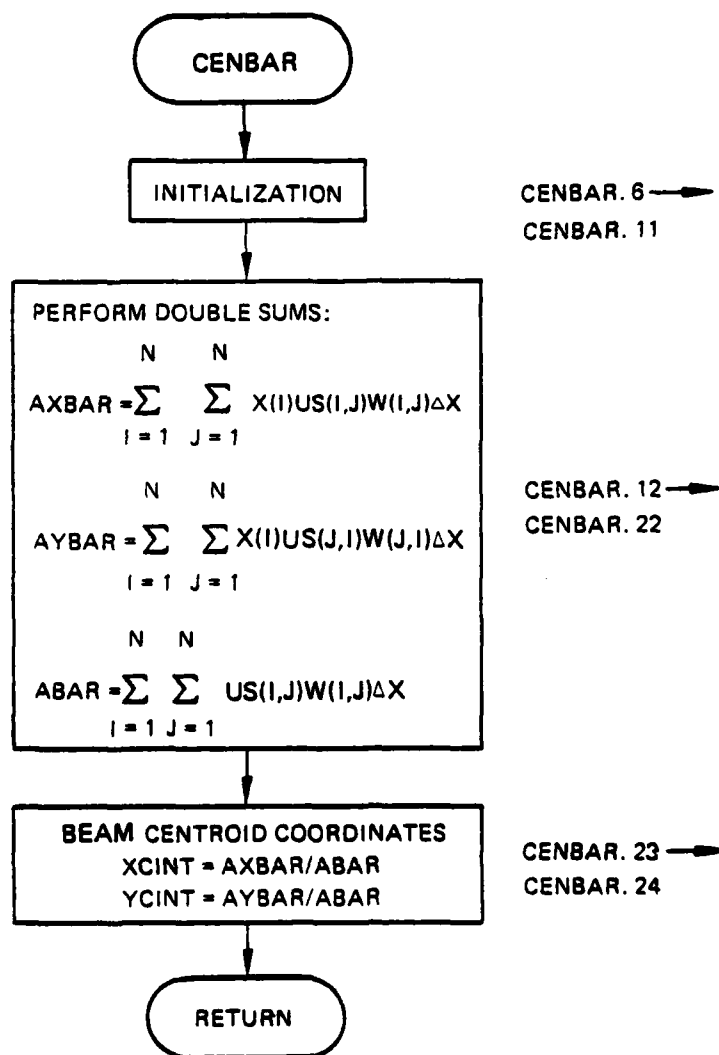


Figure 20. Subroutine CENBAR organization.

b. Formalism -- Let $E(x,y)$ represent the field and let $w(x,y)$ be a weighting function defined by

$$w(x,y) = \begin{cases} 1, & \text{if } |E(x,y)|^2 > 0.1 (|E|_{\max}^2) \\ 0, & \text{if } |E(x,y)|^2 \leq 0.1 (|E|_{\max}^2) \end{cases} \quad (37)$$

Then the intensity-weighted centroid coordinates are found from

$$\vec{x}_c = \frac{\iint dx dy |E(x,y)|^2 w(x,y) \vec{x}}{\iint dx dy |E(x,y)|^2 w(x,y)} \quad (38)$$

where the integrals are numerically evaluated over the calculation region.

c. Fortran

Argument List

NPTS = Number of points in x direction

DX = spacing between two adjacent points

X = coordinate array

US = intensity array = $|CU(I)|^2 = |E(x,y)|^2$

XCINT = Centroid coordinate in the X direction } \vec{x}_c

YCINT = Centroid coordinate in the Y direction }

UMAX = Maximum Intersity

The incoming parameters are NPTS,DX,X,US,UMAX. They are unchanged by this routine and are used to calculate XCINT and YCINT.

Note: The subroutine assumes that the field is square. Computer printout of subroutine CENBAR follows.

SUBROUTINE CENBAR 76/175 OPT=1 FIN 4.6+452 04/27/79 12.23.47

	SUBROUTINE CENBAR (NPTS, DX, X, US, XCINT, YCINT, UMAX)	CENBAR	2
C	CENTROID LOCATION MODEL	CENBAR	3
C	THIS ROUTINE LOCATES THE INTENSITY WEIGHTED CENTROID OF THE	CENBAR	4
C	COMPLEX FIELD	CENBAR	5
	LEVEL 2: NPTS, X, US	CENBAR	6
	DIMENSION X(1), US(1)	CENBAR	7
	AYBAR=0.	CENBAR	8
	UCUF = .1 * UMAX	CENBAR	9
	AYBAR=0.	CENBAR	10

ABAR=0.	CENBAH	11
DO 10 I=1,NPTS	CENBAH	12
ADY=0.	CENBAH	13
ADA=0.	CENBAH	14
DO 11 J=1,NPTS	CENBAH	15
IJ = I*(J-1)*NPTS	CENBAH	16
J1 = J*(I-1)*NPTS	CENBAH	17
IF (US(IJ) .GT. UCUT) AUX = AUX + US(IJ)	CENBAH	18
11 IF (US(J1) .GT. UCUT) ADY = ADY + US(J1)	CENBAH	19
AABAR=AXBAR*AUX*OAX(I)	CENBAH	20
AYBAR=AYBAR*ADY*OAX(I)	CENBAH	21
10 ABAR=ABAR*AUX*OAX	CENBAH	22
XCINT=AABAR/ABAR	CENBAH	23
YCINT=AYBAR/ABAR	CENBAH	24
RETURN	CENBAH	25
END	CENBAH	26

9. SUBROUTINE DENSITY

Called from: CAVITY.

Calls: LINTERP, ROSN, ROSN6

a. Purpose -- This routine controls the generation of the cavity density-induced phase distortion for each cavity in the optical train. DENSITY provides a choice of density fields including interpreted test data from several devices and the ability to read in density fields from tape. Little formal calculation is done within the routine itself, other than the generation of multipliers and certain other constants used by the interpolation routines. DENSITY does tabulate spline coefficients if any are used to generate the phase distorting field, and provides a decile plot of the phase field. Figure 21 shows the subroutine DENSITY flow chart.

Argument List

FLAG	flag for density field selection
IF	file number where MOD 6 density field is stored
IN	file number where input card data is stored
NPX	number of cavity density grid points in X direction
NPY	number of cavity density grid points in Y direction
NSYM	flag for symmetry of field
RHO	free stream static density
XLEN	X-dimension (flow direction) of cavity segment
YLEN	Y-dimension (sidewall-to-sidewall) of cavity segment
ZSLAB	Z-dimension (optical direction) of cavity segment

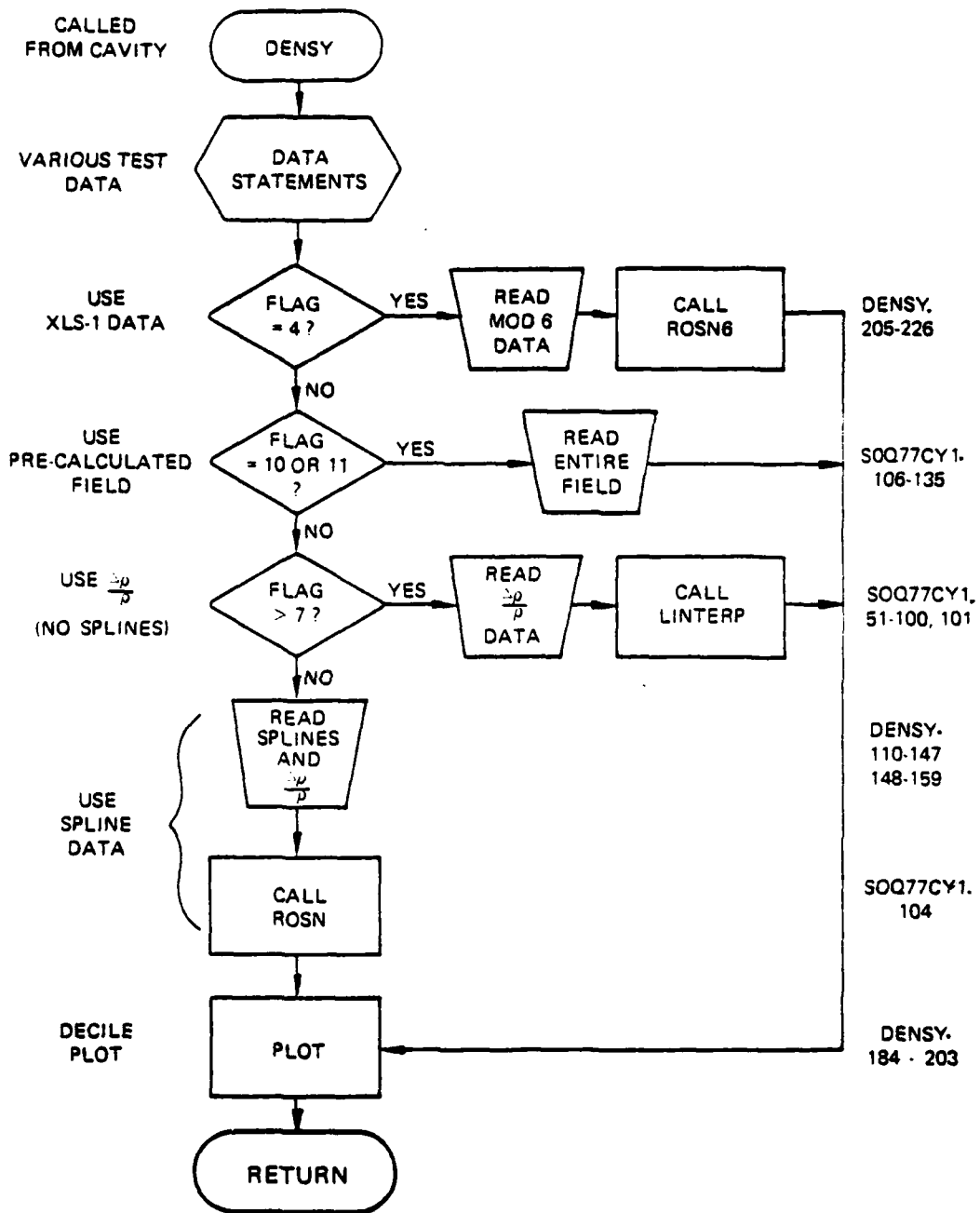


Figure 21. Subroutine DENSITY flow chart.

Commons Modified

/MELT/

Variables Modified

P	storage array for density induced phase distribution
X4	} spline coefficient and other data useful in generation of MOD 6 (XLS-1) density field - not used for other field options
Y4	
Z4	
C4	
M4	
N4	
ROCL	

/LENSY/

Variables Modified

D	spline coefficient array
H	cavity width (sidewall-to-sidewall)
LL	flag for cavity wall symmetry
M	number of data points in spline arrays
RHOCL	centerline density variation
TITLE	field identified
TM	tangent of Mach angle
XLS	spline array center deviation from NEP
XMULT	magnifier for entire density field
Y	position array
Z	density change array

b. Relevant formalism -- Most of the formal calculations involving spline fitting a density field and interpolating the results are done external to DENSITY (see subroutines LINTERP, ROSN, and ROSN6). This routine directs the activities that generate the desired field. These activities are summarized below:

- (1) The density field is read in directly from information generated by another program and written to disk (FLAG = 10 or 11)

- (2) The sidewall density variations, but not the coefficients for a spline fit, are read in by NAMELIST or from data statements. The complete density field is generated by projecting these data into the flow along Mach lines, and linearly interpolating via LINTERP. (FLAG = 8, 8.1, 9, 9.1)
- (3) The sidewall density variations and their spline fit coefficients are read in on cards or taken from DATA statements. The complete density field is generated by interpolating with the spline fit along the projection. (FLAG = 1 through 7)

A decile plot of the density-induced optical path variation (in cm) is generated after returning from one of these actions.

Subroutine DENSITY computer printouts follow.

SUBROUTINE DENSITY 76/176 OPT=1 FIN 4.6+452 04/27/79 12.25.47

```

SUBROUTINE DENSITY(FLAG,RMU,XLEN,YLEN,ZSLAB,NPX,NPY,IP,IN,NSYM)      DENSITY      2
C THIS PROGRAM COMPUTES PHASE VARIATION IN EACH SEGMENT DUE TO      DENSITY      3
C VARIATIONS IN THE GAS DENSITY IN THE OPTICAL CAVITY. INPUT PARAMETERS DENSITY      4
C ARE:                                                                DENSITY      5
C   RMU = FREE STREAM STATIC DENSITY                                DENSITY      6
C   XLEN,YLEN,ZSLAB ARE DIMENSIONS OF SEGMENT                       DENSITY      7
C   NPX,NPY ARE NUMBER OF GRID POINTS IN X,Y DIMENSIONS           DENSITY      8
C   IF IS THE FILE ON WHICH THE MOD 6 DENSITY FIELD IS STORED      DENSITY      9
C   FLAG = FLAG FOR DENSITY FIELD SELECTION                          DENSITY     10
C   = 1. FOR CUTOFFED SIDEWALL,T=3 SEC                               DENSITY     11
C   = 2. FOR FLAT SIDEWALL,T=3 SEC                                   DENSITY     12
C   = 3. LATEST AND GREATEST TWO STAGE DENSITY FIELD               DENSITY     13
C   = 4. FOR XLS=1 MOD 6 NOZZLES NORTH AND SOUTH SIDE             DENSITY     14
C   = 5. FOR INPUT FROM CARDS OF SPLINE CO-EFFS.                   DENSITY     15
C   = 6. FOR INPUT FROM HEAD IN PREVIOUS CAVITY DEFINITION         DENSITY     16
C   =8. RUN 112 AT T=1.6 SEC RIGHT STAGE BOTH WALLS                SUQ77CY1    13
C   =8.1 HEAD NAMELIST DENS8 FOR RIGHT STAGE                       SUQ77CY1    14
C   =9. RUN 109 AT T=1.6 SEC LEFT STAGE BOTH WALLS                SUQ77CY1    15
C   =9.1 HEAD NAMELIST DENS9 FOR LEFT STAGE                         SUQ77CY1    16
C   =10. HEAD DENSITY FIELD FROM UNIT 30                            SUQ77CY1    17
C   =11. HEAD DENSITY FIELD FROM UNIT 31                             SUQ77CY1    18
C                                                                      DENSITY     17
C   IMPLICIT COMPLEX(C)                                             DENSITY     18
C   LEVEL 2, P                                                       DENSITY     19
C   REAL C4                                                           DENSITY     20
C   EQUIVALENCE (M4,M)                                              DENSITY     21
C   COMMON /MELT/ P(16384),                                          DENSITY     22
X     A*(21),Y*(21*81),Z*(21*81),C*(21*81),M*(21),N*,NUCL        DENSITY     23
X     A,DUMYS(4*39*)                                                CUHR2       5
C   DIMENSION TITLE1(20),TITLE2(20),                                DENSITY     25
X     Y1(50),Z1(50),U1(50),Y2(45),Z2(45),U2(45),                  DENSITY     26
X     Y3(50),Z3(50),U3(50),TITLE3(20)                               DENSITY     27
X     ,TITLE8(20),Y8(50),Z8(50),U8(50)                              SUQ77CY1    19
X     ,Y8W(50),Z8W(50),U8W(50)                                       SUQ77CY1    20
X     ,Y9(50),Z9(50),U9(50)                                       SUQ77CY1    21
X     ,Y9W(50),Z9W(50),U9W(50),TITLE9(20)                          SUQ77CY1    22
C   DIMENSION TLE(12),IP(190)                                        CUHR1       50
C   COMMON/LENS/Y(51,2),Z(51,2),U(51,2),TM(2),XLS(2),M,XMULT(2),  DENSITY     29

```

```

X HMOCL(2),M(2),TITLE(20),LL
NAMELIST /UENSB/ TM8,M8,AM8,M8W,Y8,Z8,Y8W,Z8W
NAMELIST /UENS9/ TM9,M9,AM9,M9W,Y9,Z9,Y9W,Z9W
DATA GDC /0.228/
DATA TM3/.21034/, M3/50/, AM3/.01/
DATA Y1/-5.,-4.,-3.,-2.3,-2.2,-2.1,-2.05,-2.,-1.95,-1.90,-1.85,
A-1.8,-1.7,-1.35,-1.,-.9,-.8,-.7,-.65,-.6,-.5,-.45,-.4,-.35,-.30,
B-.25,-.2,-.15,-.1,-.05,0.,.05,.1,.15,.2,.25,.3,.35,.4,.5,.6,.65,
C.7.,.75,.85,1.,2.,3.,4.,5./
DATA Z1/50.004,.0035,.003,.002,.001,.0005,700.,.001,.002,.007,.014
A..017,.015,.006,-.006,-.016,-.016,-.01,-.004,.001,20.002,.004,.012
B..018,.0215,.022,.022,.021,.017,.013,.0115,.011,.0105,60.01/
DATA U1/ 20.2086020E-3,-.1043010E-2,.4768047E-2,-.6898764E-1,
A-.2881728E-1,-.2891209 ,-.1469915E-1,.3479176 ,-.1769713 ,
B.3599678 ,-.6290010E-1 ,.8716343E-2,-.4442123E-2 ,.9052100E-2,
C-.0592148E-1 ,.2546337 ,-.3526136 ,.2806414E+1,-.1273043E+1,
U .0154237 ,-.2344455E+1,-.3218101E+1,-.1578136E+1 ,.2330648E+1,
E .1855542E+1 ,.4667183E+1,-.1244275E+1 ,.3299169 ,-.2475342E+1,
F-.2834512E-1 ,.1887734 ,.4073251E+1,-.2081778E+1,-.5461377 ,
G-.1733670E+1 ,.2808207 ,-.5846124 ,-.3223710 ,.6191941E-1,
M .7469374E-1 ,.6274989 ,-.1866843 ,.1187585 ,.3706905E-1,
I-.2735865E-2 ,.7321330E-3,-.1926665E-3 ,.3853331E-4,-.3853331E-4/
DATA Y2/-5.,-4.,-3.5,-3.,-2.75,-2.5,-2.,-1.75,-1.5,-1.25,
A-1.1,-1.,-.95,-.9,-.85,-.8,-.75,-.7,-.65,-.6,-.55,-.5,-.45,-.4,
B0.,.05,.1,.15,.2,.25,.3,.4,.5,.6,.7,.8,.9,1.,1.1,1.15,2.,2.5,5./
DATA Z2/40.025,-.022,-.019,-.013,-.008,30.004,-.0035,-.002,-.001
A5.400.,.003,.006,-.002,-.01,-.02,-.022,-.013,-.001,.01,.017,.019,
B.015,.007,.002,0.,-.002,-.004,-.006,30.008,-.006,-.003,.006,.016,
C.026,.016/
DATA U2/20.1920707E-2,-.1536566E-1,.5954193E-1,-.3852028E-1,
A .9453919E-1,-.5163651E-1 ,.1600686E-1,-.1083904 ,.3355691E-1,
B-.2583672E-1 ,.2152010 ,-.3372499 ,.9930979 ,-.1235141E+1,
C .3474687 ,-.1547332 ,.2999660 ,.3096702 ,-.1538646E+1,
D-.7550822 ,-.1992212E+1 ,.3923432E+1 ,.5496481E+1 ,.4901429 ,
E-.2570528 ,-.1861931E+1,-.1849221E+1,-.2557182E+1,-.2276048E+1,
F .2061374E+1 ,.1230549E+1 ,.2164280 ,-.6455882E-1 ,.4180724E-1,
G-.1026701 ,.3688734 ,-.1728234 ,.3224205 ,.8314146E-1,
H-.5498634E-1 ,.4180500E-2,-.1060723E-2 ,.6239550E-4 ,.6239550E-4/
DATA TITLE1/ 4MFLAT,4M SIU,4MEWAL,4ML UE,4MNSIT,4MY FI,4MELD ,
14MASE,4MD UN,4M SR,4MI DA,4MTA ,6*4M /
DATA TITLE2/ 4M C,4MONTU,4MUREU,4M SIU,4MEWAL,4ML UE,4MNSIT,
14MY FI,4MELD ,4MHASE,4MD UN,4M SR,4MI DA,4MTA ,6*4M /
DATA TITLE3/ 4M LAT,4MEST ,4MTWO ,4M STA,4MGE D,4MNSI,4MTY F,
14MELD,4M LAM,4MINAR,4M - 2,4M PER,4MCLNT,7*4M /
DATA Y3/ -3.5466666, -2.4355555, -1.3244444, -0.4355555, -0.3244444,
X -0.257777, -0.235555, -0.224444, -0.220000, -0.215555,
X -0.213333, -0.206666, -0.202222, -0.191111, -0.180000,
X -0.175555, -0.171111, -0.164444, -0.157777, -0.146666,
X -0.124444, -0.113333, -0.102222, -0.091111, -0.080000,
X -0.068888, -0.057777, -0.035555, -0.013333, -0.002222,
X 0.008888, 0.015555, 0.019999, 0.024444, 0.031111,
X 0.042222, 0.053333, 0.064444, 0.075555, 0.120000,
X 0.164444, 0.231111, 0.342222, 0.453333, 0.675555,
X 0.897777, 1.342222, 2.008888, 2.453333, 3.120000/
DATA Z3/ 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.02, 0.06,
X 0.13, .185, 0.55, 0.87, 1.50, 1.97, 2.04, 2.06, 2.04,
X 1.98, 1.85, 1.50, 1.27, 1.15, 1.10, 1.07, 0.98, 0.88,
X 0.67, 0.47, 0.34, 0.16, 0.09, 0.05, 0.03, 0.01, 0.00,
X 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
X 0.00, 0.00, 0.00, 0.00, 0.00/
DATA U3/ 20.291422E-02,-.145711E-01, .619272E-01,-.998122E+00,
X .522010E+01,-.387664E+02, .119415E+04, .145780E+04, .208712E+04,
X .886166E+04, .267319E+04,-.337096E+04, .107416E+03,-.483470E+04,
X-.226811E+04,-.128032E+04,-.970171E+03,-.238490E+03,-.111126E+03,
X-.640623E+03, .139299E+04, .414640E+03, .350441E+03,-.844406E+03,
X .111185E+03,-.063363E+02, .819162E+02,-.119828E+03,-.902860E+03,
X .130126E+04,-.305338E+03, .159978E+04,-.188176E+02, .346199E+03,
X .374916E+02,-.100062E+02, .257350E+01,-.287744E+00, .759831E-01,
X-.161884E-01, .330596E-02,-.886024E-03, .158135E-03,-.413956E-04,
X .744651E-05,-.164175E-05, .508161E-06,-.781786E-07,-.781786E-07/
DATA TITLE4/4M ,4MNUM ,4M112 ,4M1.0 ,4MSEC ,4MHIGH,
SUQ77CY1 25

```

```

X 4MT ST,4MAGE ,4MEAST,4M ANU,4MWEST,4M WAL,4ML AN,4MALYI,4MIC , SUQ77CY1 26
X 4M ,4M ,4M ,4M ,4M ,4M SUQ77CY1 27
DATA TM8/.20345/M8/22/.XMB/.0093/. M8W/22/ SUQ77CY1 28
DATA Y8/-1.8=-1.7,-1.6=-1.5=-1.4=-1.3=-1.2=-1.1=-.8=-.7=-.6=-.5. SUQ77CY1 29

X -.4=-.3=-.2=-.1=-.04.0=.06=.1=.2=.37/ SUQ77CY1 30
DATA Z8/U...3=.8=1.1=1.2=1.8=2.4=2.8=3.1=3.6=3.9=3.8=3.3=3.8=3.4. SUQ77CY1 31
X 3.1=8.3=8.1=7.6=7.7=10.4=13.42/ SUQ77CY1 32
DATA Y8W/-2...-1.9=-1.7=-1.6=-1.5=-1.4=-1.3=-1.2=-1.1=-.8=-.7. SUQ77CY1 33
X -.6=-.5=-.4=-.3=-.2=-.06=0...1=.2=.36/ SUQ77CY1 34
DATA Z8W/U...5=.8=1.1=1.1=.8=.1=-.4=-.9=-1.5=-1.7=-1.8=-1.7. SUQ77CY1 35
X -1.2=.3=2.3=2.4=3.5=6..7.8=10./ SUQ77CY1 36
DATA TITLE9/4M ,4MUN ,4MIU9 ,4MI.8 ,4MSEC ,4MLEFI, SUQ77CY1 37
X 4M ST,4MAGE ,4MEAST,4M ANU,4MWEST,4M WAL,4ML AN,4MALYI,4MIC , SUQ77CY1 38
X 4M ,4M ,4M ,4M ,4M ,4M SUQ77CY1 39
DATA TM9/.20345/M9/16/.XMY/.01/. M9W/19/ SUQ77CY1 40
DATA Y9/-1.76=-1.28=-1.2=-1.18=-.86=-.64=-.52=-.42=-.36=-.2. SUQ77CY1 41
X -.12=-.06=.08=.16=.42=.52/ SUQ77CY1 42
DATA Z9/-3.1=1.1=1.1=1.1=-.8=-1.0=-1.05=-1.3=-.6=1.4=1.7=1.05=0.. SUQ77CY1 43
X .27=.34=.53/ SUQ77CY1 44
DATA Y9W/-1.76=-1.52=-1.4=-1.26=-1.2=-.96=-.66=-.44=-.37=-.3. SUQ77CY1 45
X -.2=-.1=-.06=.02=.05=.1=.32=.44=.62/ SUQ77CY1 46
DATA Z9W/-3.2=.9=1.7=1.0=.35=-.7=-1.2=-1.1=-.55=1.03=2.3=2.05. SUQ77CY1 47
X .45=.3=.5=2.5=3.7=6.8/ SUQ77CY1 48
C..... DENSITY 98
DATA BLANK/4M DENSITY 99
M = YLEN DENSITY 100
XMACH=4.56 DENSITY 101
LAG=FLAG*.1 DENSITY 102
MUT = NPX*NPY DENSITY 103
DU 1629 IZERU=1,MUT DENSITY 104
1629 P(IZERU) = 0. DENSITY 105
C CALL ZERU(P(1),P(MUT)) DENSITY 106
LL=1 DENSITY 107
IF(LAG.EU.5.UR.LAG.EU.7.UR.LAG.EU.8.UR.LAG.EU.9) LL=(NSYM-2) SUQ77CY1 49
GO TO (100,200,300,...500,2,500,800,901,1001,1001).LAG SUQ77CY1 50
C CUNTOURED SIDEWALL DENSITY FIELD DENSITY 110
100 TM(1)=TM2*SQRT((XM2**2-1.)/(XMACH**2-1.)) DENSITY 111
XMULG=(XM2**2/SQRT(XM2**2-1.))/(XMACH**2/SQRT(XMACH**2-1.)) DENSITY 112
XMULT(1) = 1./XMULG DENSITY 113
XLS(1)=0.0 DENSITY 114
M(1) = 45 DENSITY 115
DU 110 I=1,45 DENSITY 116
Y(I,1)=Y2(I) DENSITY 117
Z(I,1)=Z2(I) DENSITY 118
110 O(I,1)=O2(I) DENSITY 119
DU 120 I=1,20 DENSITY 120
120 TITLE(1)=TITLE2(I) DENSITY 121
GO TO 2 DENSITY 122
C FLAT SIDEWALL DENSITY FIELD DENSITY 123
200 TM(1)=TM1*SQRT((XM1**2-1.)/(XMACH**2-1.)) DENSITY 124
XMULG=(XM1**2/SQRT(XM1**2-1.))/(XMACH**2/SQRT(XMACH**2-1.)) DENSITY 125
XMULT(1) = 1./XMULG DENSITY 126
XLS(1)=0.0 DENSITY 127
M(1) = 50 DENSITY 128
DU 210 I=1,50 DENSITY 129
Y(I,1)=Y1(I) DENSITY 130
Z(I,1)=Z1(I) DENSITY 131
210 O(I,1)=O1(I) DENSITY 132
DU 220 I=1,20 DENSITY 133
220 TITLE(1)=TITLE1(I) DENSITY 134
GO TO 2 DENSITY 135
C LATEST AND GREATEST TWO STAGE DENSITY FIELD DENSITY 136
300 TM(1) = TM3 DENSITY 137
XLS(1)=0.0 DENSITY 138
XMULT(1) = XM3 DENSITY 139
M(1) = M3 DENSITY 140
DU 310 I = 1,M3 DENSITY 141
Y(I,1) = Y3(I) DENSITY 142
Z(I,1) = Z3(I) DENSITY 143
310 O(I,1) = O3(I) DENSITY 144

```

DO 320 I = 1,20	UENSY	145
J20 TITLE(1) = TITLE3(1)	UENSY	146
GO TO 2	UENSY	147
C***** ALL STAGE DENSITY FIELD (ANALYTICAL SIDEWALL PROJECTION) *****	SQU77CY1	51
800 IF (FLAG.LT.8.05) GO TO 802	SQU77CY1	52
HEAD (5,UENSB)	SQU77CY1	53
HEAD (5,807) TITLE8	SQU77CY1	54
807 FUMMAT(20A4)	SQU77CY1	55
802 TM(1) = TMB	SQU77CY1	56
XSEED=7.	SQU77CY1	57
XLS(1) = 0.0	SQU77CY1	58
XMULT(1) = XMB	SQU77CY1	59
M(1) = MB	SQU77CY1	60
DO 810 I=1,MB	SQU77CY1	61
Y(I,1) = YB(I)	SQU77CY1	62
Z(I,1) = ZB(I)	SQU77CY1	63
810 O(I,1) = 0.0	SQU77CY1	64
IF(LL.EQ.1) GO TO 815	SQU77CY1	65
TM(2) = TMB	SQU77CY1	66
XLS(2) = 0.0	SQU77CY1	67
XMULT(2) = XMB	SQU77CY1	68
M(2) = MB	SQU77CY1	69
DO 811 I=1,MB	SQU77CY1	70
Y(I,2) = YB(I)	SQU77CY1	71
Z(I,2) = ZB(I)	SQU77CY1	72
811 O(I,2) = 0.0	SQU77CY1	73
815 DO 820 I=1,20	SQU77CY1	74
820 TITLE(1) = TITLE8(1)	SQU77CY1	75
GO TO 2	SQU77CY1	76
901 IF (FLAG.LT.9.05) GO TO 904	SQU77CY1	77
HEAD (5,UENSB)	SQU77CY1	78
HEAD (5,807) TITLE9	SQU77CY1	79
904 TM(1) = TMB	SQU77CY1	80
XSEED=7.	SQU77CY1	81
XLS(1) = 0.0	SQU77CY1	82
XMULT(1) = XMB	SQU77CY1	83
M(1) = MB	SQU77CY1	84
DO 910 I=1,MB	SQU77CY1	85
Y(I,1) = YB(I)	SQU77CY1	86
Z(I,1) = ZB(I)	SQU77CY1	87
910 O(I,1) = 0.0	SQU77CY1	88
IF(LL.EQ.1) GO TO 915	SQU77CY1	89
TM(2) = TMB	SQU77CY1	90
XLS(2) = 0.0	SQU77CY1	91
XMULT(2) = XMB	SQU77CY1	92
M(2) = MB	SQU77CY1	93
DO 911 I=1,MB	SQU77CY1	94
Y(I,2) = YB(I)	SQU77CY1	95
Z(I,2) = ZB(I)	SQU77CY1	96
911 O(I,2) = 0.0	SQU77CY1	97
915 DO 920 I=1,20	SQU77CY1	98
920 TITLE(1) = TITLE9(1)	SQU77CY1	99
GO TO 2	SQU77CY1	100
500 READ (IN,987) (TITLE(I),I=1,17)	UENSY	148
507 FUMMAT (17A4)	UENSY	149
DO 705 I = 1,3	UENSY	150
705 TITLE(17*I) = BLANK	UENSY	151
709 FUMMAT (JF10.6,15)	UENSY	152
709 FUMMAT (ZF10.6,13.0)	UENSY	153
2 DO 503 L = 1,LL	UENSY	154
IF (LAG.NE.5.AND.LAG.NE.7) GO TO 222	UENSY	155
HEAD (IN,989) XLS(L), XMULT(L), TM(L), M(L)	UENSY	156
MMM = M(L)	UENSY	157
DO 502 I = 1,MMM	UENSY	158
502 HEAD (IN,988) Y(I,L),Z(I,L),O(I,L)	UENSY	159
C COMPUTE PHASE DISTRIBUTION IN SEGMENT	UENSY	160
222 WRITE(6,56) (TITLE(I),I=1,20)	UENSY	161
56 FUMMAT(1M1,2X,20A4)	UENSY	162
WRITE(6,3) NMO, M, FLAG,XLS(L),XMULT(L),TM(L),M(L)	UENSY	163
3 FUMMAT(50MU NMO M FLAG XLS XMULT TM	UENSY	164
1M /E10.3,5A,F7.3,7X,F5.1,2P6.3,F8.5,13/17X,1MS,11X,6MDELNMO,6X.	UENSY	165

211MCUEFFICIENT)	
MM = M(L)	DENSITY 166
WRITE(6,*) (Y(I,L),Z(I,L),U(I,L),I=1,MM)	DENSITY 167
* FOMAT(1UX,10,5,5X,10,5,4X,14,7)	DENSITY 168
503 HNUCL(L)=-HMU*GUC*ZSLAB*XMULT(L)	DENSITY 169
OX=XLEN/NPX	DENSITY 170
OY=YLEN/NPY/(NSYM*1)	DENSITY 171
IZ=0	DENSITY 172
DO 10 I=1,NPY	DENSITY 173
S=OY*(I-5)	DENSITY 174
DO 10 J=1,NPX	DENSITY 175
X=OX*(J-5)	DENSITY 176
IZ=IZ+1	DENSITY 177
IF(LAG.EQ.8.OR.LAG.EQ.9)CALL LINTERP(X,S,UP)	DENSITY 178
C IF(X.GT.20.)WRITE(6,2051)X,S,UP,IZ	SOU77CY1 101
2051 FOMAT(10X,9HX 5 UP IZ,3(5X,15.7),15)	SOU77CY1 102
IF(LAG.LT.8) CALL HUSN(X,S,UP)	SOU77CY1 103
10 P(IZ)=UP	SOU77CY1 104
C	DENSITY 180
GO TO 1000	DENSITY 181
C ((MODIFIED 1/14/77 FAA TO READ 2 DENSITY FIELDS FROM DISK	SOU77CY1 105
C FLAG=10. HEADS FIELD FROM UNIT 30	SOU77CY1 106
C FLAG=11. HEADS FIELD FROM UNIT 31	SOU77CY1 107
1001 IF(LAG.EQ.10)IDENS=30	SOU77CY1 108
IF(LAG.EQ.11)IDENS=31	SOU77CY1 109
C))	SOU77CY1 110
NUB = MUT	SOU77CY1 111
NUBB = NUB	SOU77CY1 112
IF(NSYM.NE.0)WRITE(6,113)	SOU77CY1 113
113 FOMAT(5X,4JHEHUR=DENSITY FIELD CHOSEN NOT COMMENSURATE ,	SOU77CY1 114
A45M WITH SYMMETRIC MESH. PROGRAM STOP ENCOUNTERED //)	SOU77CY1 115
IF(NSYM.NE.0)STOP	SOU77CY1 116
IF(NUBB.NE.0)WRITE(6,112)	SOU77CY1 117
IF(NUBB.NE.0)STOP	SOU77CY1 118
112 FOMAT(5X,39MCURRENT MESH PTS NOT IN AGREEMENT WITH	SOU77CY1 120
X,49MSTORED DENSITY VALUES. PROGRAM STOP IN DENSITY.PLZ	SOU77CY1 121
Y,11MCHECK INPUT /)	SOU77CY1 122
PHASE = (ZSLAB/198.32)	SOU77CY1 123
HEAD(IDENS)(P(IZ),IZ=1,NUBB)	SOU77CY1 124
C ((REVISED ON OR BEFORE 12/7/76 P. ADAMEK	SOU77CY1 125
HEAD(IDENS)AVOPU	SOU77CY1 126
HE=INU IDENS	SOU77CY1 127
WRITE(6,1986)AVOPU	SOU77CY1 128
1986 FOMAT(34M AVOPU) IN AREA OF CONVEX MIRROR = (E15.6)	SOU77CY1 129
DO 955 KK = 1,NUBB	SOU77CY1 130
M(KK) = (M(KK) - AVOPU) * PHASE	SOU77CY1 131
955 CONTINUE	SOU77CY1 132
WRITE(6,114)IDENS,NUBB	SOU77CY1 133
114 FOMAT(5X,20MDENSITY FIELD HEAD FROM UNIT ,13,2M, .15,	SOU77CY1 134
ARMPTS HEAD //)	SOU77CY1 135
1000 CONTINUE	SOU77CY1 136
C --PLOTPT=0. FOR NO PLOTTING POINTS	SOU77CY1 137
C --PLOTPT=1. FOR PLOTTING POINTS IN X DIR. THRU CENTER OF CAVITY	SOU77CY1 138
C --PLOTPT=2. FOR PLOTTING POINTS IN Y DIR. THRU CENTER OF BEAM	SOU77CY1 139
PLOTPT=0.	SOU77CY1 140
IF(PLOTPT.EQ.0.) GO TO 1236	SOU77CY1 141
WRITE(6,1987)	SOU77CY1 142
1987 FOMAT(//20X,15MPLOTTING POINTS	SOU77CY1 143
III=0	SOU77CY1 144
AUX=XLEN/NPX	SOU77CY1 145
XUY=YLEN/NPY/(NSYM*1)	SOU77CY1 146
DO 1235 I=1,NPY	SOU77CY1 147
YYY=XUY*(I-5)	SOU77CY1 148
DO 1235 J=1,NPX	SOU77CY1 149
XXX=AUX*(J-5)	SOU77CY1 150
III=III+1	SOU77CY1 151
IF(PLOTPT.EQ.1.) GO TO 1989	SOU77CY1 152
IF(XXX.LE.5.0.OR.XXX.GE.5.0//)GO TO 1235	SOU77CY1 153
GO TO 1490	SOU77CY1 154
1989 IF(YYY.LE.5.0.OR.YYY.GE.5.0//)GO TO 1235	SOU77CY1 155
1990 WRITE(6,1456)XXX,YYY,M(III)	SOU77CY1 156
1456 FOMAT(1UX,1/MX , Y , UPU/566 ,3(1E15.7,5X))	SOU77CY1 157

1235 CONTINUE	SOU77CY1	158
C))	SOU77CY1	159
1236 CONTINUE	SOU77CY1	160
C UHA0 PICTURE OF PHASE SHIFT PER SEGMENT	DENSY	184
PMAX=P(1)	DENSY	185
PMIN=P(1)	DENSY	186
DO 50 I=1,NUT	DENSY	187
PMIN = AMIN(PMIN,P(I))	DENSY	188
50 PMAX = AMAX(PMAX,P(I))	DENSY	189
UP=PMAX-PMIN	DENSY	190
IF (LAG.LT.10) WRITE (6,51) TITLE	SOU77CY1	161
IF (LAG.GE.10) WRITE (6,52)3	SOU77CY1	162
5203 FORMAT(1M)	SOU77CY1	163
51 FORMAT(1M1.25X.20A4)	DENSY	192
INTE=1	DENSY	193
IF (NMX.GT.128) INTE=2	DENSY	194
DO 52 J=1,NPY	DENSY	195
KJ= (NPY-J) * NMX	DENSY	196
DO 53 I=1,NPX*INTE	DENSY	197
53 IP(I)=10.0*(1.0-(I(1-KJ)-PMIN)/UP)	DENSY	198
52 WRITE(6,54) (IP(I),I=1,NPX*INTE)	DENSY	199
54 FORMAT(2X.130I1)	DENSY	200
WRITE(6,55) PMIN,PMAX,UP	DENSY	201
55 FORMAT(13HMIN VALUE IS,E15.7,5X,12HMAX VALUE IS,E15.7,5X,	DENSY	202
13H NORMALIZING FACTOR FOR ABOVE PLOT IS ,E15.7)	DENSY	203
RETURN	DENSY	204
C XLS=1 MOD 6 NOZZLES SPLINE DATA FROM FILE 14	DENSY	205
400 HEAD(IF,1400) TITLE	DENSY	206
1400 FORMAT(12A4)	DENSY	207
HEAD(IF,1401) N4,M4	DENSY	208
1401 FORMAT(16I5)	DENSY	209
HEAD(IF,1402) X4,Y4,Z4,C4	DENSY	210
1402 FORMAT(5E16.8)	DENSY	211
DO 401 I=1,12	DENSY	212
401 TITLE(I)=TITLE(I)	DENSY	213
DO 402 I=13,20	DENSY	214
402 TITLE(I)=BLANK	DENSY	215
K=M4(I)	DENSY	216
M=Y4(1,K)-Y4(1,1)	DENSY	217
HUCL=-HMU*GUC*ZSLAB	DENSY	218
DX=XLEN/NPX	DENSY	219
DY=YLEN/(NPY*(NSYM+1))	DENSY	220
DO 410 I=1,NPX	DENSY	221
X=DX*(I-.5)	DENSY	222
DO 410 J=1,NPY	DENSY	223
S= DY*(J-.5) - (YLEN-D.) / 2.	DENSY	224
CALL HUSN6(X,S,UP)	DENSY	225
410 P(1+(J-1)*NPX) = UP	DENSY	226
GO TO 1000	SOU77CY1	164
END	DENSY	228

10. SUBROUTINE FOURT

a. Purpose -- Subroutine FOURT performs a forward or backward Fast Fourier Transform on any multidimensional complex array by efficiently performing the summation.

$$A_m = \sum_{N=0}^{N-1} X_n e^{\pm 2\pi imn/N} \quad (39)$$

The transform pair that needs to be evaluated is

$$F(s) = \int_{-\infty}^{\infty} f(x) e^{2\pi i x s} dx \quad (40)$$

and

$$f(x) = \int_{-\infty}^{\infty} F(s) e^{-2\pi i x s} ds \quad (41)$$

To digitally evaluate an integral, the continuous form of an integral must be changed to its discrete form. For example,

$$G = \int_a^b g(x) dx \Rightarrow \lim_{N \rightarrow \infty} \sum_{n=0}^N g_n \Delta X \quad (42)$$

b. Relevant formalism -- Assume that all the intervals, ΔX_n , are chosen to be equal and that the infinite sum can be approximated by a finite sum. Then,

$$G = \sum_{n=0}^{N-1} g_n (x_{n+1} - x_n) \text{ with } g_n = g \left(x = \frac{n(b-a)}{N} \right) \quad (43)$$

$$G = \sum_{n=0}^{N-1} g_n \left[(n+1) \frac{(b-a)}{N} - n \frac{(b-a)}{N} \right]$$

or

$$G = \Delta X \sum_{n=0}^{N-1} g_n \int_a^b g(x) dx \quad (44)$$

To evaluate Equations (40) and (41) by the approximate form (Eq. (44)), assume that the function $f(x)$ is spatially bounded in $0 \leq x < 2L$ and that it is a band-limited function so that $F(s)$ is confined in the region $-B \leq S \leq B$. To perform either a backward or forward Fourier transform, the functions f and F should differ in form only by the sign of the exponent. Therefore, the properties of F must be evaluated so that its region can be changed to $0 \leq S \leq 2B$. This is easily done by replicating the function $f(x)$ so that it is periodic with period $2L$. This will not change the value of f in the region of interest and, by proper choice of N , will return the desired function F .

A sampled function, f_s , can be analytically represented by a Dirac delta function:

$$f_s(x) = \sum_{n=0}^{N-1} f_n \delta(x - n\Delta x) \text{ with } \Delta x = \frac{2L}{N} \quad (45)$$

A replicated function can be represented by a convolution:

$$\begin{aligned} f_{\text{rep}}(x) &= \int_0^{2L} dx' f(x') \sum_{n=-\infty}^{\infty} \delta(x - (x' + 2LN)) \\ &= f(x) \circ \sum_{n=-\infty}^{\infty} \delta(x - n2L) \end{aligned} \quad (46)$$

Therefore, a sampled and replicated function is represented by:

$$\hat{f}(x) = \sum_{n=0}^{N-1} f_n \delta(x - n\Delta x) \circ \sum_{m=-\infty}^{\infty} \delta(x - nN\Delta x) \quad (47)$$

The Fourier Transform $\hat{F}(s)$ of $\hat{f}(x)$ is

$$\hat{F}(s) = F\{\hat{f}\} = F\left\{\sum_{n=0}^{N-1} f_n \delta(x - n\Delta x)\right\} F\left\{\sum_{m=-\infty}^{\infty} \delta(x - nN\Delta x)\right\} \quad (48)$$

by the convolution theorem. Since

$$\sum_{n=-\infty}^{\infty} \delta(x-na) = \frac{1}{a} \sum_{n=-\infty}^{\infty} e^{2\pi i n \frac{x}{a}} \quad (49)$$

one finds,

$$\hat{F}(s) = \sum_{n=0}^{N-1} f_n e^{2\pi i s n \Delta x} \sum_{m=-\infty}^{\infty} \frac{1}{N \Delta x} \delta\left(s - \frac{n}{N \Delta x}\right) \quad (50)$$

Rearranged this gives

$$\hat{F}(s) = \frac{1}{N \Delta x} \sum_{m=-\infty}^{\infty} \delta\left(s - \frac{m}{N \Delta x}\right) \sum_{n=0}^{N-1} f_n e^{2\pi i m n / N} \quad (51)$$

Recalling Equations (40) and (44), define

$$F_n = \Delta x \sum_{n=0}^{N-1} f_n e^{2\pi i n m / N} = F_{n+N} \quad (52)$$

Then

$$\hat{F}(s) = \frac{1}{N(\Delta x)^2} \sum_{m=-\infty}^{\infty} F_m \delta\left(s - \frac{m}{N \Delta x}\right) \quad (53)$$

Since $F_m = F_{m+N}$, one can rewrite the above as a replication for every N point.

$$\hat{F}(s) = \frac{1}{N(\Delta x)^2} \sum_{m=0}^{N-1} F_m \delta\left(s - \frac{m}{N \Delta x}\right) \sum_{n=-\infty}^{\infty} \delta\left(s - \frac{n}{\Delta x}\right) \quad (54)$$

Therefore, by replicating $f(x)$ with period $2L$, F is periodic with period $1/\Delta x$.

So by choosing N so that $N/2L \geq 2B$, rewrite the limits for F as $0 \leq \underline{S} < B$.

Since

$$\delta_{nk} = \frac{1}{N} \sum_{m=0}^{N-1} e^{2\pi i m(n-k)/N} = \begin{cases} 1, & n = k \\ 0, & n \neq k \end{cases} \quad (55)$$

invert (52) to find

$$f_n = \frac{1}{N\Delta x} \sum_{m=0}^{N-1} F_m e^{-2\pi i mn/N} \quad (56)$$

Thus, choosing $\Delta s = 1/N\Delta x$, the transform pair becomes

$$F_m = \Delta x \sum_{n=0}^{N-1} f_n e^{-2\pi i mn/N} \quad (57)$$

$$f_n = \Delta s \sum_{m=0}^{N-1} F_m e^{-2\pi i mn/N} \quad (\Delta x \Delta s = \frac{1}{N}) \quad (58)$$

where, with $N/2L \geq 2B$, F_m represents $F(s)$ for $0 \leq \underline{S}_m < 2B$ ($\underline{S}_m = m\Delta s$) and f_n represents $f(x)$ for $0 \leq \underline{x}_n < 2L$ ($\underline{x}_n = n\Delta x$).

The transform pair f_n and F_m are now in a form usable by the Fast Fourier Transform (FFT). The FFT evaluates the sum

$$A_r = \sum_{k=0}^{N-1} X_k e^{\pm 2\pi i rk/N} \quad (59)$$

Following Higgins (Ref. 9), this sum can be split into two sums (choosing the + sign in the exponent):

$$A_r = \sum_{\substack{k=0 \\ \text{(keven)}}}^{N-1} X_k e^{\pi i r k / N} + \sum_{\substack{k=0 \\ \text{(kodd)}}}^{N-1} X_k e^{2\pi i r k / N} \quad (60)$$

Let

$$k = 0, 1, 3, 5, \dots \frac{N}{2} - 1 \quad (61)$$

then

$$A_r = \sum_{k=0}^{\frac{N}{2}-1} \left[y_k e^{2\pi i r 2k / N} + z_k e^{2\pi i r (2k+1) / N} \right] \quad (62)$$

Letting

$$B_r \equiv \sum_{k=0}^{\frac{N}{2}-1} y_k e^{4\pi i r k / N} \quad (63)$$

and

$$C_r \equiv \sum_{k=0}^{\frac{N}{2}-1} z_k e^{4\pi i r k / N} \quad (64)$$

9. Wiggins, R.J., "Fast Fourier Transform: An Introduction With Some Minicomputer Experiments," AJP, 44, 1976.

A_r can be written

$$A_r = B_r + C_r e^{2\pi i r/N} \quad (65)$$

Define

$$W_n \equiv e^{2\pi i/N} \quad (66)$$

Then,

$$A_r = B_r + C_r + (W_n)^r \quad (67)$$

By letting $r \rightarrow r + N/2$:

$$A\left(r + \frac{N}{2}\right) = B_r - (W_n)^r C_r \quad (68)$$

Therefore, A_r can be evaluated by doing two sums, each containing $N/2$ terms. However, these sums need to be performed for only half the r 's ($0 \leq r < \frac{N}{2}$) since $A_{r + N/2}$ is found using the two sums used in the evaluation of A_r . By

initially forcing N to be a power of two by completing the array to be transformed with zeros, continue to divide each successful sum into two, until a "sum" is reduced to just one number, taking care to note that N changes with each division. When using the FFT, care must be taken to scale the output correctly since the FFT evaluates only sums of the form

$$A_r = \sum_{n=0}^{N-1} x_n e^{\pm 2\pi i n r/N} \quad (69)$$

and as can be seen from Equations (58) the Fourier Transforms contain Δx or Δs : If only forward then backward transforming is done, it is sufficient to divide the final answer by N for each dimension as is indicated by the last part of Equation (58).

Note that when the data are returned from the FFT the first data point is either the $x = 0$ or the $s = 0$ point. To see the actual frequency space pictures, assume a two-dimensional case. An isointensity printer plot of FFT output in frequency space might look like that shown in Figure 22.

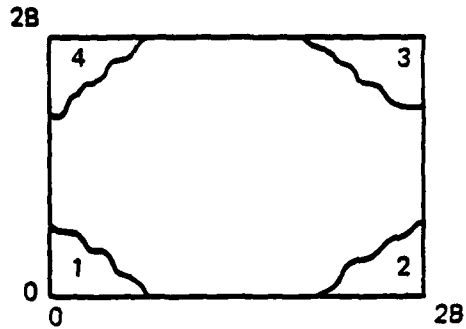


Figure 22. Example of isointensity printer plot of FFT output in frequency space.

To see the $-B$ to $+B$ version, the adjacent cells shown in Figure 23 must be added to Figure 22.

The subroutine FOURT computer printouts follow.

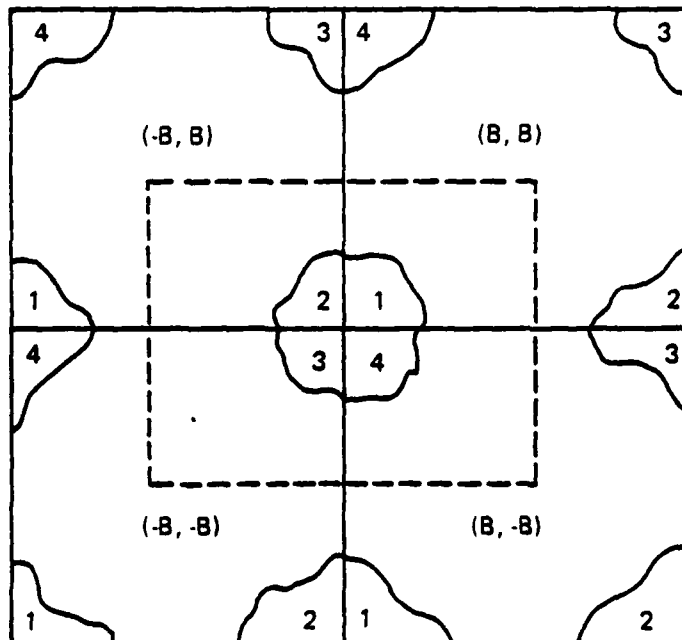


Figure 23. $-B$ to $+B$ version of isointensity printer plot of FFT output in frequency space.


```

SUBROUTINE FOURT(DATA,NAH,NN,ISIGN)
C.....
C THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN
C TRANSFORM(K1,K2,...) = SUM(DATA(J1,J2,...)*EXP(ISIGN*2*PI*(SQRT(-1)
C *(J1-1)*(K1-1)/NN(1)+(J2-1)*(K2-1)/NN(2)+...))), SUMMED FOR ALL
C J1, K1 BETWEEN 1 AND NN(1), J2, K2 BETWEEN 1 AND NN(2), ETC.
C THERE IS NO LIMIT TO THE NUMBER OF SUBSCRIPTS. DATA IS A
C MULTIDIMENSIONAL COMPLEX ARRAY WHOSE REAL AND IMAGINARY
C PARTS ARE ADJACENT IN STORAGE, SUCH AS FORTRAN IV PLACES THEM.
C IF ALL IMAGINARY PARTS ARE ZERO (DATA ARE DISGUISED REAL), SET
C IFORM TO ZERO TO CUT THE RUNNING TIME BY UP TO FORTY PERCENT.
C OTHERWISE, IFORM = +1. THE LENGTHS OF ALL DIMENSIONS ARE
C STORED IN ARRAY NN, OF LENGTH NDIM. THEY MAY BE ANY POSITIVE
C INTEGERS, WHO THE PROGRAM RUNS FASTER ON COMPOSITE INTEGERS, AND
C ESPECIALLY FAST ON NUMBERS HIGH IN FACTORS OF TWO. ISIGN IS +1
C OR -1. IF A -1 TRANSFORM IS FOLLOWED BY A +1 ONE (OR A +1
C BY A -1) THE ORIGINAL DATA REAPPEAR, MULTIPLIED BY NTOT (=NN(1)*
C NN(2)*...). TRANSFORM VALUES ARE ALWAYS COMPLEX, AND ARE RETURNED
C IN ARRAY DATA, REPLACING THE INPUT. IN ADDITION, IF ALL
C DIMENSIONS ARE NOT POWERS OF TWO, ARRAY WORK MUST BE SUPPLIED.
C COMPLEX OF LENGTH EQUAL TO THE LARGEST NON 2**K DIMENSION.
C OTHERWISE, REPLACE WORK BY ZERO IN THE CALLING SEQUENCE.
C NORMAL FORTRAN DATA ORDERING IS EXPECTED, FIRST SUBSCRIPT VARYING
C FASTEST. ALL SUBSCRIPTS BEGIN AT ONE.
LEVEL 2, DATA
DIMENSION DATA(NAH),NN(2),IFACT(32),WORK(100)
NDIM=2
IFORM=+1
NI=1.00
NH=1.00
WSTPR=1.00
WSTPI=1.00
FWOP=6.283185307
IF(NDIM-1)920,1.1
1 NTOT=2
DO 2 IDIM=1,NDIM
IF(NN(IDIM))920,920,2
2 NTOT=NTOT*NN(IDIM)
NP1=2
DO 910 IDIM=1,NDIM
N=NN(IDIM)
NP2=NP1*N
IF(N-1)920,900,5
5 MN=
NT=0=NP1
IF=1
IDIV=2
10 IQUOT=M/IDIV
IHEM=M-IDIV*IQUOT
IF(IQUOT-IDIV)50,11,11
11 IF(IHEM)20,12,20
12 NT=0=NT*0=N*FWO
M=IQUOT
GO TO 10
20 IDIV=3
30 IQUOT=M/IDIV
IHEM=M-IDIV*IQUOT
IF(IQUOT-IDIV)60,31,31
31 IF(IHEM)40,32,40
32 IFACT(IF)=IDIV
IF=IF+1
M=IQUOT
GO TO 30
40 IDIV=IDIV*2
GO TO 30
50 IF(IHEM)60,51,60
51 NT=0=NT*0=N*FWO
GO TO 70
FUORT 2
FUORT 3
FUORT 4
FUORT 5
FUORT 6
FUORT 7
FUORT 8
FUORT 9
FUORT 10
FUORT 11
FUORT 12
FUORT 13
FUORT 14
FUORT 15
FUORT 16
FUORT 17
FUORT 18
FUORT 19
FUORT 20
FUORT 21
FUORT 22
FUORT 23
FUORT 24
FUORT 25
FUORT 26
FUORT 27
FUORT 28
FUORT 29
FUORT 30
FUORT 31
FUORT 32
FUORT 33
FUORT 34
FUORT 35
FUORT 36
FUORT 37
FUORT 38
FUORT 39
FUORT 40
FUORT 41
FUORT 42
FUORT 43
FUORT 44
FUORT 45
FUORT 46
FUORT 47
FUORT 48
FUORT 49
FUORT 50
FUORT 51
FUORT 52
FUORT 53
FUORT 54
FUORT 55
FUORT 56
FUORT 57
FUORT 58
FUORT 59
FUORT 60
FUORT 61
FUORT 62
FUORT 63
FUORT 64
FUORT 65
FUORT 66
FUORT 67
FUORT 68
FUORT 69

```

60	IFACT(IF)=M	FOURT	70
70	NUN2=NP1*(NP2/NTWU)	FOURT	71
	ICASE=1	FOURT	72
	IF(IDIM=4)71,90,90	FOURT	73
71	IF(IFURM)72,72,90	FOURT	74
72	ICASE=2	FOURT	75
	IF(IDIM=1)73,73,90	FOURT	76
73	ICASE=3	FOURT	77
	IF(NTWU=NP1)90,90,74	FOURT	78
74	ICASE=4	FOURT	79
	NTWU=NTWU/2	FOURT	80
	N=N/2	FOURT	81
	NP2=NP2/2	FOURT	82
	NTOT=NTOT/2	FOURT	83
	I=3	FOURT	84
	DO 40 J=2,NTOT	FOURT	85
	DATA(J)=DATA(I)	FOURT	86
80	I=I+2	FOURT	87
90	I1HNG=NP1	FOURT	88
	IF(ICASE=2)100,95,100	FOURT	89
95	I1HNG=NP0*(1+NPNEV/2)	FOURT	90
100	IF(NTWU=NP1)600,600,110	FOURT	91
110	NP2MF=NP2/2	FOURT	92
	J=1	FOURT	93
	DO 150 I2=1,NP2,NUN2	FOURT	94
	IF(J=I2)120,130,130	FOURT	95
120	I1MAX=I2-NUN2-2	FOURT	96
	DO 125 I1=I2,I1MAX,2	FOURT	97
	DO 125 I3=I1,NTOT,NP2	FOURT	98
	J3=J+I3-12	FOURT	99
	TEMPH=DATA(I3)	FOURT	100
	TEMPI=DATA(I3+1)	FOURT	101
	DATA(I3)=DATA(J3)	FOURT	102
	DATA(I3+1)=DATA(J3+1)	FOURT	103
	DATA(J3)=TEMPH	FOURT	104
125	DATA(J3+1)=TEMPI	FOURT	105
130	M=NP2MF	FOURT	106
140	IF(J=M)150,150,145	FOURT	107
145	J=J-M	FOURT	108
	M=M/2	FOURT	109
	IF(M=NUN2)150,140,140	FOURT	110
150	J=J+M	FOURT	111
	NUN2T=NUN2-NUN2	FOURT	112
	IPAH=NTWU/NP1	FOURT	113
310	IF(IPAH=2)350,330,320	FOURT	114
320	IPAH=IPAH/4	FOURT	115
	GO TO 310	FOURT	116
330	DO 340 I1=1,I1HNG,2	FOURT	117
	DO 340 JJ=1,NUN2,NP1	FOURT	118
	DO 340 K1=JJ,NTOT,NUN2T	FOURT	119
	K2=K1-NUN2	FOURT	120
	TEMPH=DATA(K2)	FOURT	121
	TEMPI=DATA(K2+1)	FOURT	122
	DATA(K2)=DATA(K1)-TEMPH	FOURT	123
	DATA(K2+1)=DATA(K1+1)-TEMPI	FOURT	124
	DATA(K1)=DATA(K1)+TEMPH	FOURT	125
340	DATA(K1+1)=DATA(K1+1)+TEMPI	FOURT	126
350	MMAX=NON2	FOURT	127
360	IF(MMAX=NP2MF)370,600,600	FOURT	128
370	LMAX=MAXU(NUN2T,MMAX/2)	FOURT	129
	IF(MMAX=NON2)+05,+05,380	FOURT	130
380	THETA=(NON1*FLUAT(NUN2)/FLUAT(4*MMAX)	FOURT	131
	IF(15IGN)400,390,390	FOURT	132
390	THETA=-THETA	FOURT	133
400	WH=COS(THETA)	FOURT	134
	WI=SIN(THETA)	FOURT	135
	WSPH=-2.*WH*WI	FOURT	136
	WSPI=2.*WH*WI	FOURT	137
405	DO 570 L=NUN2,LMAX,NUN2T	FOURT	138
	M=L	FOURT	139
	IF(MMAX=NON2)+20,+20,+10	FOURT	140

410	W2H=WH*WH-W1*W1	FOURT	141
	W2I=2.*WH*W1	FOURT	142
	W3H=W2H*WH-W2I*W1	FOURT	143
	W3I=W2H*W1+W2I*WH	FOURT	144
420	DU 530 I1=1,11RNG,2	FOURT	145
	DU 530 J3=1,1,NUN2,NP1	FOURT	146
	KMIN=J3*IPAN*H	FOURT	147
	IF (HMAX-NUN2) 430,430,440	FOURT	148
430	KMIN=J3	FOURT	149
440	KDIF=IPAN*HMAX	FOURT	150
450	KSTEP=4*KDIF	FOURT	151
	DU 520 K1=KMIN,NTOT,KSTEP	FOURT	152
	K2=K1*KDIF	FOURT	153
	K3=K2*KDIF	FOURT	154
	K4=K3*KDIF	FOURT	155
	IF (HMAX-NUN2) 460,460,480	FOURT	156
460	U1H=DATA(K1)*DATA(K2)	FOURT	157
	U1I=DATA(K1+1)*DATA(K2+1)	FOURT	158
	U2H=DATA(K3)*DATA(K4)	FOURT	159
	U2I=DATA(K3+1)*DATA(K4+1)	FOURT	160
	U3H=DATA(K1)-DATA(K2)	FOURT	161
	U3I=DATA(K1+1)-DATA(K2+1)	FOURT	162
	IF (ISIGN) 470,475,475	FOURT	163
470	U4H=DATA(K3+1)-DATA(K4+1)	FOURT	164
	U4I=DATA(K4)-DATA(K3)	FOURT	165
	GO TO 510	FOURT	166
475	U4H=DATA(K4+1)-DATA(K3+1)	FOURT	167
	U4I=DATA(K3)-DATA(K4)	FOURT	168
	GO TO 510	FOURT	169
480	T2H=W2H*DATA(K2)-W2I*DATA(K2+1)	FOURT	170
	T2I=W2H*DATA(K2+1)+W2I*DATA(K2)	FOURT	171
	T3H=WH*DATA(K3)-W1*DATA(K3+1)	FOURT	172
	T3I=WH*DATA(K3+1)+W1*DATA(K3)	FOURT	173
	T4H=W3H*DATA(K4)-W3I*DATA(K4+1)	FOURT	174
	T4I=W3H*DATA(K4+1)+W3I*DATA(K4)	FOURT	175
	U1H=DATA(K1)+T2H	FOURT	176
	U1I=DATA(K1+1)+T2I	FOURT	177
	U2H=T3H+T4H	FOURT	178
	U2I=T3I+T4I	FOURT	179
	U3H=DATA(K1)-T2H	FOURT	180
	U3I=DATA(K1+1)-T2I	FOURT	181
	IF (ISIGN) 490,500,500	FOURT	182
490	U4H=T3I-T4I	FOURT	183
	U4I=T4H-T3H	FOURT	184
	GO TO 510	FOURT	185
500	U4H=T4I-T3I	FOURT	186
	U4I=T3H-T4H	FOURT	187
510	DATA(K1)=U1H+U2H	FOURT	188
	DATA(K1+1)=U1I+U2I	FOURT	189
	DATA(K2)=U3H+U4H	FOURT	190
	DATA(K2+1)=U3I+U4I	FOURT	191
	DATA(K3)=U1H-U2H	FOURT	192
	DATA(K3+1)=U1I-U2I	FOURT	193
	DATA(K4)=U3H-U4H	FOURT	194
	DATA(K4+1)=U3I-U4I	FOURT	195
520	KMIN=4*(KMIN-J3)+J3	FOURT	196
	KDIF=KSTEP	FOURT	197
	IF (KDIF-NP2) 450,530,530	FOURT	198
530	CUNFINUE	FOURT	199
	HMAX=H	FOURT	200
	IF (ISIGN) 540,550,550	FOURT	201
540	TEMPH=WH	FOURT	202
	WH=H1	FOURT	203
	H1=TEMPH	FOURT	204
	GO TO 560	FOURT	205
550	TEMPH=WH	FOURT	206
	WH=H1	FOURT	207
	H1=TEMPH	FOURT	208
560	IF (H-LMAX) 565,565,410	FOURT	209
565	TEMPH=WH	FOURT	210
	WH=WH*WSIPH-W1*WSTP1*WH	FOURT	211

570	W[1]=W[1]*SIPH+TEMPH*WSTP[1]	FOUNT	212
	IPAR=J-IPAR	FOUNT	213
	MMAI=MMAI+MMAI	FOUNT	214
	GO TO 360	FOUNT	215
600	IF (INTW0=NP2) 605,700,700	FOUNT	216
605	IF1=NUNZ	FOUNT	217
	IF=1	FOUNT	218
	NP1M=NP1/2	FOUNT	219
610	IFP2=[FP1]/FACT(IF)	FOUNT	220
	J1NNG=NP2	FOUNT	221
	IF (ICASE=3) 612,611,612	FOUNT	222
611	J1NNG=(NP2+IFP1)/2	FOUNT	223
	J2STP=NP2/FACT(IF)	FOUNT	224
	J1NG2=(J2STP+IFP2)/2	FOUNT	225
612	J2MIN=1+IFP2	FOUNT	226
	IF (IFP1=NP2) 615,600,600	FOUNT	227
615	DO 630 J2=J2MIN,IFP1,IFP2	FOUNT	228
	THEFA=-T*UP1*FLUAT(J2-1)/FLUAT(NP2)	FOUNT	229
	IF (ISIGN) 625,620,620	FOUNT	230
620	THEFA=-THEFA	FOUNT	231
625	SINTH=SIN(THEFA/2.)	FOUNT	232
	WSTPH=-2.*SINTH*SINTH	FOUNT	233
	WSTP1=SIN(THEFA)	FOUNT	234
	WR=WSTPH*1.	FOUNT	235
	WI=WSTP1	FOUNT	236
	J1MIN=J2+IFP1	FOUNT	237
	DO 630 J1=J1MIN,J1NNG,IFP1	FOUNT	238
	I1MAX=J1+I1NNG-2	FOUNT	239
	DO 630 I1=J1,I1MAX,2	FOUNT	240
	DO 630 J3=1,NTOT,NP2	FOUNT	241
	J3MAX=J3+IFP2-NP1	FOUNT	242
	DO 630 J3=J3,J3MAX,NP1	FOUNT	243
	TEMPH=DATA(J3)	FOUNT	244
	DATA(J3)=DATA(J3)*WR+DATA(J3+1)*WI	FOUNT	245
630	DATA(J3+1)=TEMPH*WI+DATA(J3+1)*WR	FOUNT	246
	TEMPH=WR	FOUNT	247
	WR=WR*WSTPH+W[1]*WSTP1+WR	FOUNT	248
635	W[1]=TEMPH*WSTP1+W[1]*WSTPH+W[1]	FOUNT	249
640	THEFA=-T*UP1/FLUAT(FACT(IF))	FOUNT	250
	IF (ISIGN) 650,645,645	FOUNT	251
645	THEFA=-THEFA	FOUNT	252
650	SINTH=SIN(THEFA/2.)	FOUNT	253
	WSTPH=-2.*SINTH*SINTH	FOUNT	254
	WSTP1=SIN(THEFA)	FOUNT	255
	KSTEP=2*W[1]/FACT(IF)	FOUNT	256
	KHANG=KSTEP*(FACT(IF)/2)*1	FOUNT	257
	DO 698 I1=1,I1NNG-2	FOUNT	258
	DO 698 J3=1,NTOT,NP2	FOUNT	259
	DO 690 KMIN=1,KHANG,KSTEP	FOUNT	260
	J1MAX=J3+J1NNG-IFP1	FOUNT	261
	DO 680 J1=J3,J1MAX,IFP1	FOUNT	262
	J3MAX=J1+IFP2-NP1	FOUNT	263
	DO 680 J3=J1,J3MAX,NP1	FOUNT	264
	J2MAX=J3+IFP1-IFP2	FOUNT	265
	K=KMIN+(J3-J1+(J1-J3)/FACT(IF))/NP1M	FOUNT	266
	IF (KMIN=1) 655,655,655	FOUNT	267
655	SUMR=0.	FOUNT	268
	SUMI=0.	FOUNT	269
	DO 660 J2=J3,J2MAX,IFP2	FOUNT	270
	SUMR=SUMR+DATA(J2)	FOUNT	271
660	SUMI=SUMI+DATA(J2+1)	FOUNT	272
	WORK(K)=SUMR	FOUNT	273
	WORK(K+1)=SUMI	FOUNT	274
	GO TO 680	FOUNT	275
665	KCONJ=K+2*(N-KMIN+1)	FOUNT	276
	J2=J2MAX	FOUNT	277
	SUMR=DATA(J2)	FOUNT	278
	SUMI=DATA(J2+1)	FOUNT	279
	ULDSH=0.	FOUNT	280
	ULDSI=0.	FOUNT	281
	J2=J2-IFP2	FOUNT	282

670	TEMPH=SUMH	FOURT	283
	TEMPI=SUMI	FOURT	284
	SUMH=(W0WH*SUMH-OLUSH+DATA(J2)	FOURT	285
	SUMI=(W0WH*SUMI-OLUSI+DATA(J2+1)	FOURT	286
	OLUSH=TEMPH	FOURT	287
	OLUSI=TEMPI	FOURT	288
	J2=J2-IFP2	FOURT	289
	IF(J2-J3)675,675,670	FOURT	290
675	TEMPH=WH*SUMH+OLUSH+DATA(J2)	FOURT	291
	TEMPI=WI*SUMI	FOURT	292
	W0HK(K)=TEMPH-TEMPI	FOURT	293
	W0HK(KCUNJ)=TEMPH+TEMPI	FOURT	294
	TEMPH=WH*SUMI+OLUSI+DATA(J2+1)	FOURT	295
	TEMPI=WI*SUMH	FOURT	296
	W0HK(K+1)=TEMPH+TEMPI	FOURT	297
	W0HK(KCUNJ+1)=TEMPH-TEMPI	FOURT	298
680	CONTINUE	FOURT	299
	IF(KMIN-1)685,685,686	FOURT	300
685	WH=WSTPH+1.	FOURT	301
	WI=WSTPI	FOURT	302
	GO TO 690	FOURT	303
686	TEMPH=WH	FOURT	304
	WH=WH+WSTPH-WI*WSTPI+WH	FOURT	305
	WI=TEMPH+WSTPI-WI*WSTPH+WI	FOURT	306
690	TW0WH=WR+WH	FOURT	307
	IF(ICASE=3)692,691,692	FOURT	308
691	IF(IFP1-NP2)695,692,692	FOURT	309
692	K=1	FOURT	310
	I2MAX=I3+NP2-NP1	FOURT	311
	DO 693 I2=I3,I2MAX,NP1	FOURT	312
	DATA(I2)=W0HK(K)	FOURT	313
	DATA(I2+1)=W0HK(K+1)	FOURT	314
693	K=K+2	FOURT	315
	GO TO 698	FOURT	316
695	J3MAX=I3+IFP2-NP1	FOURT	317
	DO 697 J3=I3,J3MAX,NP1	FOURT	318
	J2MAX=J3+NP2-J2STP	FOURT	319
	DO 697 J2=J3,J2MAX,J2STP	FOURT	320
	J1MAX=J2+J1MG2-IFP2	FOURT	321
	J1CNU=J3+J2MAX+J2STP-J2	FOURT	322
	DO 697 J1=J2,J1MAX,IFP2	FOURT	323
	K=J+J-1	FOURT	324
	DATA(J1)=W0HK(K)	FOURT	325
	DATA(J1+1)=W0HK(K+1)	FOURT	326
	IF(J1-J2)697,697,696	FOURT	327
696	DATA(J1CNU)=W0HK(K)	FOURT	328
	DATA(J1CNU+1)=W0HK(K+1)	FOURT	329
697	J1CNU=J1CNU-IFP2	FOURT	330
698	CONTINUE	FOURT	331
	IF=IF+1	FOURT	332
	IFH=IFP2	FOURT	333
	IF(IFP1-NP1)/00,700,610	FOURT	334
700	GO TO (900,800,900,701),ICASE	FOURT	335
701	NHALF=N	FOURT	336
	N=N+N	FOURT	337
	THETA=-TW0H1/FLUAT(N)	FOURT	338
	IF(ISIGN)703,702,702	FOURT	339
702	THETA=THETA	FOURT	340
703	SINTH=SIN(THETA/2.)	FOURT	341
	WSTPH=2.*SINTH*SINTH	FOURT	342
	WSTPI=SIN(THETA)	FOURT	343
	WH=WSTPH+1.	FOURT	344
	WI=WSTPI	FOURT	345
	JMIN=J	FOURT	346
	JMIN=2*NHALF-1	FOURT	347
	GO TO 725	FOURT	348
710	J=JMIN	FOURT	349
	DO 720 I=JMIN,NTOT,NP2	FOURT	350
	SUMH=(DATA(I)+DATA(J))/2.	FOURT	351
	SUMI=(DATA(I+1)+DATA(J+1))/2.	FOURT	352
	UIFH=(DATA(I)-DATA(J))/2.	FOURT	353
	OIFI=(DATA(I+1)-DATA(J+1))/2.	FOURT	354

	TEMPH=WR*SUM1+U*U*IFH	FOURT	355
	TEMPI=WR*SUM1-U*U*IFH	FOURT	356
	DATA(I)=SUMH+TEMPH	FOURT	357
	DATA(I+1)=U*IFI+TEMPI	FOURT	358
	DATA(J)=SUMH-TEMPH	FOURT	359
	DATA(J+1)=-U*IFI+TEMPI	FOURT	360
720	J=J+NP2	FOURT	361
	IMIN=IMIN-2	FOURT	362
	JMIN=JMIN-2	FOURT	363
	TEMPH=WH	FOURT	364
	WR=WR*WSTPH-U*U*WSTPI+WR	FOURT	365
	WI=TEMPH*WSTPI-U*U*WSTPH+WI	FOURT	366
725	IF (IMIN-JMIN) 710,730,740	FOURT	367
730	IF (ISIGN) 731,740,740	FOURT	368
731	DO 735 I=IMIN,NTUT,NP2	FOURT	369
735	DATA(I+1)=-DATA(I+1)	FOURT	370
740	NP2=NP2+NP2	FOURT	371
	NTUT=NTUT+NTUT	FOURT	372
	J=NTUT+1	FOURT	373
	IMAX=NTUT/2+1	FOURT	374
745	IMIN=IMAX-2*NMALF	FOURT	375
	I=IMIN	FOURT	376
	GO TO 755	FOURT	377
750	DATA(J)=DATA(I)	FOURT	378
	DATA(J+1)=-DATA(I+1)	FOURT	379
755	I=I+2	FOURT	380
	J=J+2	FOURT	381
	IF (I-IMAX) 750,760,760	FOURT	382
760	DATA(J)=DATA(IMIN)+DATA(IMIN+1)	FOURT	383
	DATA(J+1)=0.	FOURT	384
	IF (I-J) 770,780,780	FOURT	385
765	DATA(J)=DATA(I)	FOURT	386
	DATA(J+1)=DATA(I+1)	FOURT	387
770	I=I-2	FOURT	388
	J=J-2	FOURT	389
	IF (I-IMIN) 775,775,785	FOURT	390
775	DATA(J)=DATA(IMIN)+DATA(IMIN+1)	FOURT	391
	DATA(J+1)=0.	FOURT	392
	IMAX=IMIN	FOURT	393
	GO TO 745	FOURT	394
780	DATA(1)=DATA(1)+DATA(2)	FOURT	395
	DATA(2)=0.	FOURT	396
	GO TO 800	FOURT	397
800	IF (I1HNG-NP1) 805,900,900	FOURT	398
805	DO 805 I=1,NTUT,NP2	FOURT	399
	I2MAX=I3+NP2-NP1	FOURT	400
	DO 805 I2=I3,I2MAX,NP1	FOURT	401
	IMIN=I2+I1HNG	FOURT	402
	IMAX=I2+NP1-2	FOURT	403
	JMAX=2*I3+NP1-IMIN	FOURT	404
	IF (I2-I3) 820,820,810	FOURT	405
810	JMAX=JMAX+NP2	FOURT	406
820	IF (I2-I3) 850,850,830	FOURT	407
830	J=JMAX+NP0	FOURT	408
	DO 840 I=IMIN,IMAX,2	FOURT	409
	DATA(I)=DATA(J)	FOURT	410
	DATA(I+1)=-DATA(J+1)	FOURT	411
840	J=J+2	FOURT	412
850	J=JMAX	FOURT	413
	DO 860 I=IMIN,IMAX,NP0	FOURT	414
	DATA(I)=DATA(J)	FOURT	415
	DATA(I+1)=-DATA(J+1)	FOURT	416
860	J=J+NP0	FOURT	417
900	NP0=NP1	FOURT	418
	NP1=NP2	FOURT	419
910	NPREV=NP	FOURT	420
920	COUNTINUE	FOURT	421
	RETURN	FOURT	422
	END	FOURT	423

11. SUBROUTINE FUHS

a. Purpose -- Subroutine FUHS is used to calculate the phase change due to heat release as the molecules in the lower laser level decay to the ground state, assuming supersonic flow and that the heat release has a disturbing effect (not major) on the flow. Figure 24 shows the subroutine FUHS flow chart.

b. Relevant formalism -- The equations used are based on those by Biblarz and Fuhs, (Ref. 10), and by Fuhs (Ref. 11).

Initially, it is assumed that the continuity, momentum, and energy equations for steady flow with heat addition are valid:

$$\text{Continuity: } \nabla \cdot (\rho \vec{u}) = 0 \quad (70)$$

$$\text{Momentum: } \rho \frac{D\vec{u}}{Dt} + \vec{\nabla} p = 0 \quad (71)$$

$$\text{Energy: } \nabla \cdot \rho \vec{u} \left(h + \frac{\vec{u}^2}{2} \right) = q \quad (72)$$

These are linearized, assuming

$$\rho = \rho_\infty + \rho' \quad p = p_\infty + p' \quad \vec{u} = \hat{i} (U+u') + \hat{j} v' \quad (73)$$

resulting in

$$\text{Continuity: } \rho_\infty u'_x + \rho_\infty U'_y + U \rho'_x = 0 \quad (74)$$

$$\left(u' \equiv \frac{\partial}{\partial x} u' ; \text{ etc.} \right) \quad (75)$$

$$\text{Momentum: } \left\{ \begin{array}{l} \rho_\infty U u'_x + p'_x = 0 \\ \rho_\infty U v'_x + p'_y = 0 \end{array} \right\} \quad (76)$$

10. Biblarz, O. and Fuhs, A. E., "Laser Cavity Density Changes with Kinetics of Energy Release," AIAA Journal, 12, p. 1083, August 1974.
11. Fuhs, A. E., "Quaside Area Rule for Heat Addition in Transonic and Supersonic Flight Regimes," AFAPL-TR-72-10, Air Force Propulsion Laboratory, WPAFB, Ohio, 1972.

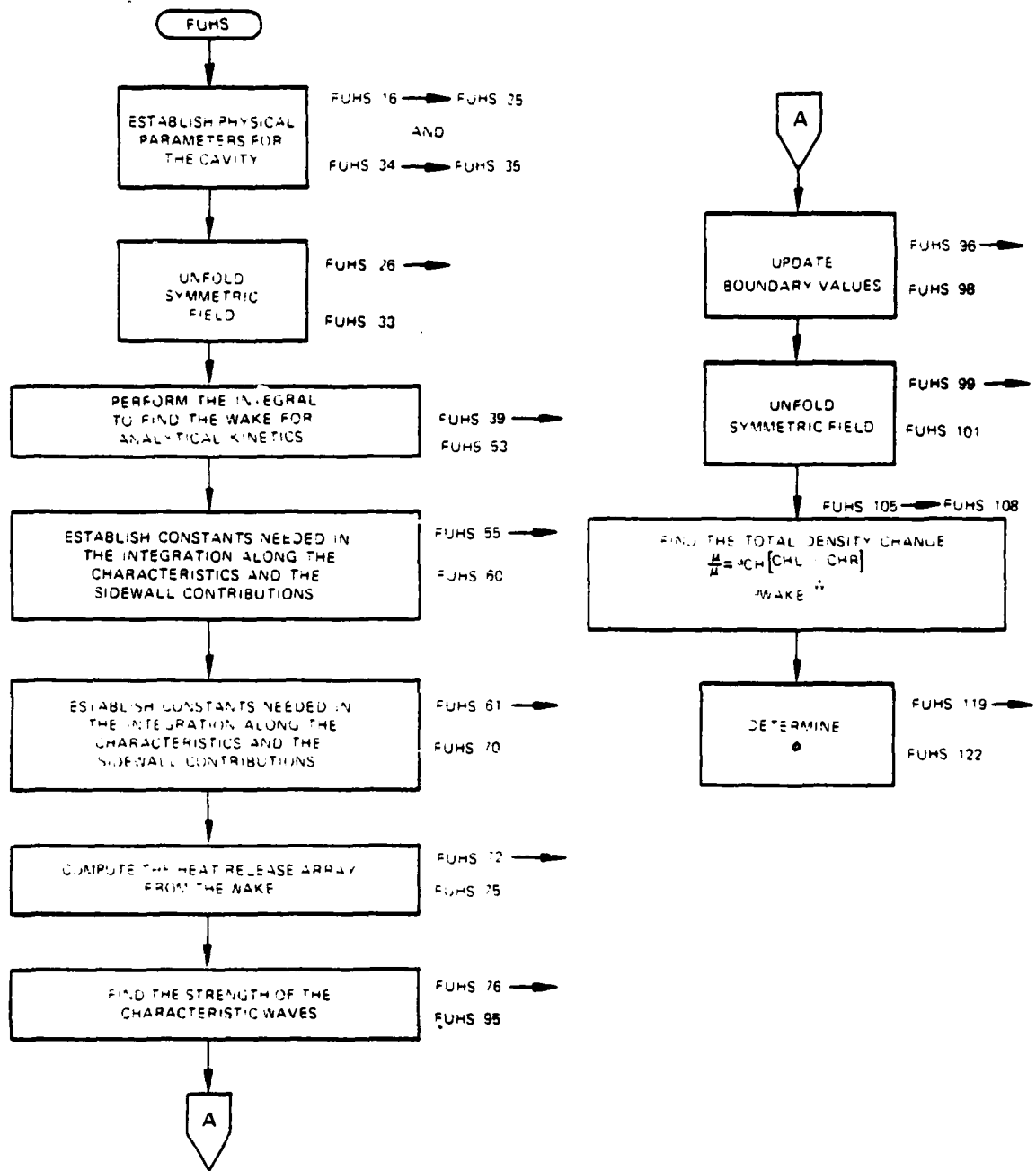


Figure 24. Subroutine FUHS organization.

$$\text{Energy: } \frac{\rho_{\infty} U_{\infty}}{\gamma - 1} \frac{\partial}{\partial} \left(\frac{P'}{\rho_{\infty}} - \gamma \rho' \right) = q \quad (77)$$

The solution is then found by using the potential for the flow as done by Tsien and Bieloch, (Ref. 12), resulting in the following equations for a heat source q in supersonic heat addition

$$u' = - \frac{(\gamma - 1)q}{2\gamma\rho\beta} \delta(x - \beta y) \quad (78)$$

$$v' = \frac{(\gamma - 1)q}{2\gamma\rho} \delta(x - \beta y) \quad (79)$$

$$P' = \frac{(\gamma - 1)qM}{2a\beta} \delta(x - \beta y) \quad (80)$$

$$\rho' = \frac{(\gamma - 1)qM}{2a^3\beta} \delta(x - \beta y) - \frac{(\gamma - 1)q}{a^2U} \delta(y) I(x) \quad (81)$$

where

$$x = \beta y \quad \text{Defines a Mach line} \quad (82)$$

$$\beta = \sqrt{M^2 - 1} \quad (83)$$

$$a = U/M \quad \text{Speed of sound} \quad (84)$$

$$I(x) = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases} \quad (85)$$

For volume heat addition $q \rightarrow dq = h(x,y)dx dy$, and the effect of all sources are added; for example,

$$u' = \frac{-(\gamma - 1)}{2\gamma\rho\beta} \iint h(x,y) dx dy \delta(x - \beta y) \quad (86)$$

$$= \frac{-(\gamma - 1)}{2\gamma\rho\beta} \int_0^s h(x = \beta y) \sin \mu ds \quad (87)$$

12. Tsien, H. E. and Milton Bielloch, "Heat Source in a Uniform Flow," Journal of the Aeronautical Sciences, December 1949, p. 746.

where the integral is taken along a streamline ($x = \beta y$) and $\sin\mu = 1/M$. S is related to x and y by

$$S = x \cos\mu \qquad S = y \sin\mu$$

The equation for density change is therefore,

$$\frac{\Delta\rho}{\rho} = \frac{1}{\rho} \left[\left(\frac{\gamma-1}{2a^3\beta} \int_0^S h(x,y) \Big|_{x=\beta y} \sin\mu ds \right) - \left(\frac{\gamma-1}{a^2 U} \iint dx'dy' h(x',y') \delta(y-y') I(x-x') \right) \right] \quad (88)$$

The first term is due to heat addition along a streamline while the second is due to the wake in the energy release region. "Heat addition in a supersonic stream causes compression waves which radiate from the heat release region. The waves reflect from the cavity walls. Downstream of the heat release region is a wake. Whereas the compression waves increase gas density the wake decreases gas density" (Ref. 12).

The heat release ($h(x,y)$ for a laser can be written:

$$h(x,y) = c \int_{x_{NEP}}^x \Delta I(x',y) \epsilon^{-(x-x')/UT} dx' \quad (89)$$

where T is the time constant for the depopulation of the lower laser level. If the depopulation were instantaneous ($T \rightarrow 0$) then the heat release would be proportional to the intensity since for every molecule emitting a photon, that same molecule gives off a quantum of heat. It has been shown (Ref. 12) that the above equation for the heat release can be used in all regions of the far cavity with only small error.

The constant C can be found by conservation of energy. Consider the following three-level molecule shown in Figure 25.

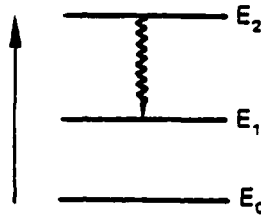


Figure 25. Three-level molecule.

The quantum efficiency η is defined as the ratio of the power out divided by the power in, so for the gain/phase segment under consideration

$$\eta = \frac{(\text{No. molecules}) (E_2 - E_1)}{(\text{No. molecules}) (E_2 - E_0)} = \frac{P}{\Delta H + \Delta P} \quad (90)$$

where

$$\Delta H = (\text{No. molecules}) (E_1 - E_0)$$

The above expression can be inverted to give

$$\Delta H = \left(\frac{1-\eta}{\eta} \right) \Delta P$$

with

$$\Delta P = \iint dx dy' \Delta I(x', y')$$

and

$$\Delta H = \iint dx dy' h(x', y') \quad (91)$$

Assume, for this calculation, that $(0,0)$ is at the corner of the sidewall and the NEP. Then,

$$\begin{aligned} \Delta H &= c \Delta z \int_0^\infty dy \int_0^\infty dx \int_0^\infty \Delta I(x', y) e^{-(x-x')/UT} dx' \\ &= c \Delta z \int_0^\infty dy \int_0^\infty dx \int_0^\infty I(x-x') \Delta I(x', y) e^{-(x-x')/UT} dx' \end{aligned} \quad (92)$$

where, recall

$$I(x-x') = \begin{cases} 1, & x > x' \\ 0 & x < x' \end{cases}$$

so

$$\begin{aligned} \Delta H &= c\Delta z \int_0^\infty dy \int_0^\infty dx \Delta I(x', y) \int_0^\infty dx I(x-x') e^{-(x-x')/UT} \\ &= c\Delta z \int_0^\infty dy \int_0^\infty dx \Delta I(x, y) \int_x^\infty dx'' e^{-x''/UT} \end{aligned} \quad (93)$$

$$\begin{aligned} \Delta H &= c\Delta z \int_0^\infty dy \int_0^\infty dx \Delta I(x, y) \left(\frac{1}{1/UT} \right) \\ &= cUT\Delta z \Delta P \end{aligned} \quad (94)$$

so

$$\frac{1-\eta}{\eta} = \frac{\Delta H}{\Delta P} = cUT\Delta z \quad (95)$$

or

$$c = \left(\frac{1-\eta}{\eta} \right) \left(\frac{1}{UT\Delta z} \right) \quad (96)$$

Since the numerical kinetics return the conditions of the wake region and not the heat addition, these must be the data used. Thus, for the analytical kinetics model, find the heat addition to the wake:

$$\begin{aligned}
W(x,y) &= \int_0^x dx' h(x',y) = c \int_0^x dx' \int_0^{x'} dx'' \Delta I(x'',y) \epsilon^{-(x'-x'')/UT} \\
&= c \int_0^\infty dx' I(x-x') \int_0^\infty dx'' I(x'-x'') \Delta I(x'',y) \epsilon^{-(x-x'')/UT} \\
&= c \int_0^\infty dx'' \Delta I(x'',y) \int_0^\infty dx' I(x-x') I(x'-x'') \epsilon^{-(x'-x'')/UT} \\
&= c \int_0^\infty dx'' \Delta I(x'',y) I(x-x'') \int_{x''}^x dx' \epsilon^{-(x'-x'')/UT} \tag{97}
\end{aligned}$$

so

$$W(x,y) = c \int_0^x dx'' \Delta I(x'',y) UT \left(1 - \epsilon^{-(x-x'')/UT} \right) \tag{98}$$

so, recalling

$$c = \frac{1-\eta}{\eta} \frac{1}{UT\Delta z} \quad \text{and} \quad \Delta I(x'',y) = 2 \left(\frac{1-G}{1+G} \right) \text{PPD from SIMPGC} \tag{99}$$

wake energy addition becomes

$$W(x,y) = \frac{2}{\Delta z} \left(\frac{1-G}{1+G} \right) \frac{1-\eta}{\eta} \int_0^x dx' \text{PPD}(x',y) \left(1 - \epsilon^{-(x-x')/UT} \right) \tag{100}$$

Now that both numerical and analytical models can give the wake integrated heat addition, the Fuhs effect is calculated in the following manner:

$$H(I,J) = \frac{1}{\Delta x} \int_{x(I-1)}^{x(I)} h(x,y) dx = \frac{W(x(I)) - W(x(I-1))}{\Delta x} \tag{101}$$

Given this average heat release function, the integral along a characteristic can be performed. Note that reflection off the sidewalls must be included, as

can be seen in Figure 26. The contribution due to reflection at P_1 is therefore found by finding the total heat released along the characteristic that reflects at P_2 , then adding this to that found along P_2P_1 .

(Note: For larger Mach angles ($>\tan^{-1}(\Delta y/2\Delta x)$), the effective number of points in the direction is multiplied by a factor of KS in the program so that only information in two mesh rectangles is needed to find heat addition at the wall, i.e., extrapolation from the two nearest the sidewall, as can be seen from the following more detailed description of how the left and right characteristic terms are found.) Assume $KS = 1$ and that the Mach angle is less than $\tan^{-1}(\Delta y/2\Delta x)$. This is assumed in the program by changing the total effective number of x coordinates to be $KS \cdot NPTS$.

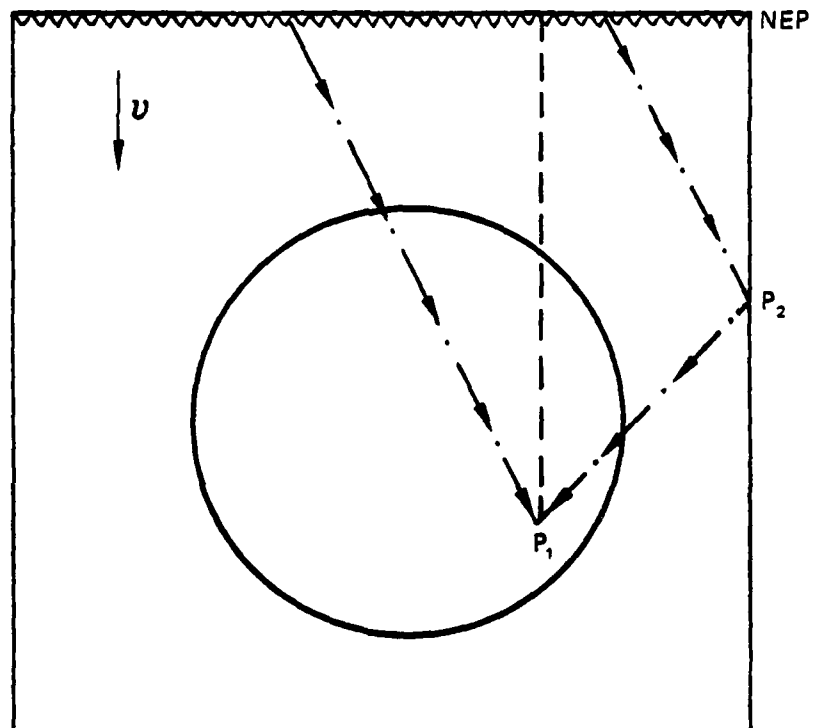


Figure 26. Average heat release function.

Consider first the left characteristic term for the (I,J) point in Figure 27:

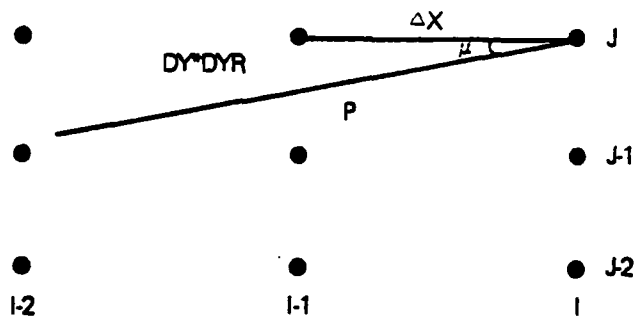


Figure 27. Left characteristic value.

The left characteristic value at (I,J) is that at P (found by a linear interpolation between the $(I-1,J)$ and $(I-1,J-1)$ points) plus the heat released in the region, again using a linear interpolation for H at $(I-1,J)$ and $(I-1,J-1)$.

Now consider a boundary point shown in Figure 28:

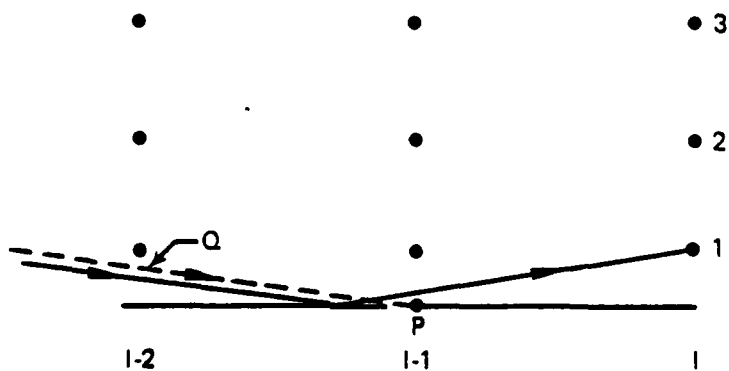


Figure 28. Boundary point.

To find the characteristic value at $(I,1)$ it is necessary to know the value at point P which is in the $(I,1)$ column on the sidewall. The value will then be a linear interpolation between the values at $(I-1,1)$ and P plus a similar linear interpolation for the added heat.

To find the characteristic value at point P, the values at (I-2,2) and (I-2,1) are extrapolated linearly toward the boundary to find the value at point Q. Heat is then added, again by linear extrapolation.

Note that this detailed analysis at the boundary assumes that the characteristic of interest lies between the boundary at (I-1) and the (I-1,1) point, hence the necessity of the restriction that $DYR = DYCH/DY$ be less than 0.5.

Analysis of the right characteristic is similar to that of the left characteristic.

The phase shift is found using the Gladstone-Dale relation.

$$n \approx 1 + C_D \quad (102)$$

The phase change $\Delta\phi$ is

$$\Delta\phi = \frac{2\pi}{\lambda} \Delta n \Delta z = \frac{2\pi}{\lambda} \left(\frac{C}{\rho_0} \Delta\rho \right) \rho_0 \Delta z \quad (103)$$

This is then added to that of the unloaded density field to establish the total phase change at the gain/phase segment.

c. Fortran

Argument List

$$\text{IIC} = \left\{ \begin{array}{l} \text{wake for numerical kinetics} \\ \Delta I \times \frac{1}{\Delta Z} \left(\frac{1-n}{\eta} \right) \text{ for analytical kinetics} \end{array} \right\}$$

DEN = phase change returned due to the FUHS effect

NCV - cavity number

Commons Changes - none

Subroutines called - none

Computer printouts of subroutine FUHS follow.


```

SUBROUTINE FUHS(ZIC,DEN,NCV)
C FUMS EFFECT ALGORITHM
C THIS ROUTINE CALCULATES THE CONTRIBUTION TO THE CAVITY DENSITY
C FIELD DUE TO STIMULATED EMISSION INDUCED HEAT ADDITION.
LEVEL 2: ZIC,DEN,AC
COMMON/CAV2/AC(5),YC(5),ZC(5),NX(5),NY(5),NS(5),XMC(5),YMC(5),
2 NGTYPE(20), SS0AIN(190,5),SATIN(5),BETA(5),RHUS(5),
3 VEL(5),GAM(5),XMACH(5),TV1(5),TV2(5),TV3(5),TVN2(5),TSCAV(5),
4 PSCAV(5),PH(5),FN2(5),FCU2(5),FM2(5),FCU(5),FU2(5),TITLE(20),
5 AVG(5), NSYM
DIMENSION ZIC( 1 ),DEN( 1 ),CHM(96,2),CHL(96,2),M(96)
ENTHP(A+B,C)=A+C*(B-A)
CALL CPUIM(15MT)
C *** CALCULATE INITIAL CONSTANTS
U = VEL(NCV)
GMA = GAM(NCV)
XMA = XMACH(NCV)
MHU = RHUS(NCV)
A = U/XMA
AHAK = (GMA-1.0)/(A**2*U*MHU)
IM=NX(NCV)
JM=NY(NCV)
DX=XC(NCV)/IM
DY=YC(NCV)/JM
IF (NSYM.EQ.0) GO TO 444
J2=JM/2
DO 445 J=1,J2
DO 445 I=1,IM
I2 = I - (J-1)*IM
I3 = I + (JM-J)*IM
ZIC(I3)=ZIC(I2)
445 DEN(I3)=DEN(I2)
444 TANMU=1.0/SQRT(XMA**2-1.0)
ACH = (GMA-1.0)*XMA/(2.0*A**3*SQRT(XMA**2-1.0)*MHU)*UY
IF(NGTYPE(NCV).EQ.1) GO TO 11
IU=IM-1
XLAG=UX/(U/BETA(NCV))
DO 15 J=1,JM
DO 14 IU=1,IU
I=IM+1-IU
N=I
SUM=J.
DO 13 IL=2,I
N=N-1
H = (I-N)*XLAG
B = 0.
IF(H.GT.20.) GO TO 12
H = 1.0/EXP(H)
12 CONTINUE
13 SUM = SUM+ZIC(N+(J-1)*IM)*(1.-B)
14 ZIC(I+(J-1)*IM)=SUM*UX
ZIC(I+(J-1)*IM) = 0.
15 CONTINUE
11 DO 6 K=1,IU
KS=K
DYCH=UX*TANMU/FLUAT(KS)
DYN=DYCH/DY
IF(DYN.LT.0.5) GO TO 7
6 CONTINUE
7 SCL=1.0*YH
DYN2=2.0*UYH
ACH=ACH*UYH
SCH=1.5*UYH
DO 1 J=1,JM
DEN(1+(J-1)*IM) = 0.
CHL(J,1)=0.
1 CHM(J,1)=0.
CHLWAL=0.
CHMWAL=0.

```

	DU 200 I=2,IM	FUMS	71
C	000 COMPUTE HEAT RELEASED AT I=1	FUMS	72
	DU 210 J=1,JM	FUMS	73
	IZ = I*(J-1)*IM	FUMS	74
	210 H(I)=ZIC(IZ)-ZIC(IZ-1)/DX	FUMS	75
C	000 COMPUTE STRENGTH OF CHARACTERISTIC WAVES	FUMS	76
	DU 100 K=1,K5	FUMS	77
	DU 50 J=1,JM	FUMS	78
C	000 LEFT RUNNING WAVE	FUMS	79
	JL=J-1	FUMS	80
	IF(J.NE.1) GO TO 20	FUMS	81
C	000 EXTRAPOLATE FOR HEAT RELEASED, USE BOUNDARY POINT	FUMS	82
	CHL(1,2)=ENTHP(CHL(1,1),CHLWAL,DYR2)*ENTHP(H(2),H(1),SCL)	FUMS	83
	GO TO 30	FUMS	84
C	000 INTERPOLATE FOR VALUE	FUMS	85
	20 CHL(J,2)=ENTHP(CHL(J,1),CHL(JL,1),DYN)*ENTHP(H(J),H(JL),DYR)	FUMS	86
C	000 RIGHT RUNNING WAVE	FUMS	87
	30 JM=J+1	FUMS	88
	IF(J.NE.JM) GO TO 40	FUMS	89
C	000 EXTRAPOLATE FOR HEAT RELEASED, USE BOUNDARY POINT	FUMS	90
	CHN(JM,2)=ENTHP(CHN(JM,1),CHNWAL,DYR2)*ENTHP(H(JL),H(JM),SCL)	FUMS	91
	GO TO 50	FUMS	92
C	000 INTERPOLATE FOR VALUE	FUMS	93
	40 CHN(J,2)=ENTHP(CHN(J,1),CHN(JM,1),DYN)*ENTHP(H(J),H(JM),DYR)	FUMS	94
	50 CONTINUE	FUMS	95
C	000 UPDATE BOUNDARY POINTS	FUMS	96
	CHLWAL=ENTHP(CHN(2,1),CHN(1,1),SCR)*ENTHP(H(2),H(1),SCR)	FUMS	97
	CHNWAL=ENTHP(CHL(JM-1,1),CHL(JM,1),SCR)*ENTHP(H(JM-1),H(JM),SCR)	FUMS	98
	DU 60 J=1,JM	FUMS	99
	CHR(J,1)=CHR(J,2)	FUMS	100
	60 CHL(J,1)=CHL(J,2)	FUMS	101
C	000 WRITE(6,03) I,H(1),CHR(1,1),CHL(1,1),CHLWAL,CHNWAL	FUMS	102
C	03 FUMMAT(1A,15,5G12.5)	FUMS	103
	100 CONTINUE	FUMS	104
C	000 GET TOTAL DENSITY CHANGE	FUMS	105
	DU 110 J=1,JM	FUMS	106
	IJ = I + (J-1)*IM	FUMS	107
	110 DEN(IJ)=ACH*(CHN(J,1)+CHL(J,1))-AWAR*ZIC(IJ)	FUMS	108
	200 CONTINUE	FUMS	109
C	000 DU 800 K=1,8	FUMS	110
C	000 WRITE(6,001)	FUMS	111
C	001 FUMMAT(1M1)	FUMS	112
C	000 IL=1+16*(K-1)	FUMS	113
C	000 IU=IL+15	FUMS	114
C	000 DU 802 J=1,JM	FUMS	115
C	002 WRITE(6,003) (DEN(I,J),I=IL,IU)	FUMS	116
C	003 FUMMAT(1A,16(2P16.3))	FUMS	117
C	000 CONTINUE	FUMS	118
	MUCL = .228*MMU*ZC(INCV)/NS(INCV)	FUMS	119
	JT = IM*JM	FUMS	120
	DU 70 J=1,JT	FUMS	121
	DEN(J) = MUCL*DEN(J)	FUMS	122
	70 CONTINUE	FUMS	123
	CALL CMUT(IM,IF,IN)	FUMS	124
	DELT=(ISKT-IF,IN)/100.	FUMS	125
	WRITE(6,778) DELT	FUMS	126
	778 FUMMAT(2UM,FUMS ANALYSIS TOOK .0125,2UM SECONDS OF CPU TIME,///)	FUMS	127
	RETURN	FUMS	128
	END	FUMS	129

12. SUBROUTINE GAINXY

a. Purpose -- GAINXY controls the gain calculations in the cavity. Figure 29 shows the Subroutine GAINXY flow chart. Either small signal gain (along one stream tube) or full-field-loaded gain is selected. From input cavity conditions (including vibrational temperatures of the constituents at

nozzle exit plane), all other thermodynamic parameters, energy levels, broadened line-width function, gain, optical cross section, and saturation intensity at a single point are given. Subroutine KINET is called to integrate the rate equations along the X-direction (streamtube). This is done only once for small signal gain. When loaded gain is selected the entire field is calculated and gain is updated by local intensity one step in the Z (propagation) direction. The loaded gain is hence a numerical (small step-wise integrated) process. This updated gain and intensity field is used to SOQ.

The single stream tube small signal gain is used in subroutine SIMPGG which computes a closed form solution of the full field loaded gain.

Subroutine MIX is called by subroutine GAINXY to calculate the transition rates.

A ratio technique is employed to effect calculation of the gain field for 9.27 μ lasing. This is triggered by GFACT = 1 for 10.60 μ ; GFACT = 1 for 9.27 μ .

b. Relevant formalism -- The option for small signal gain only or full-field loaded numerical gain is determined by IFIELD = 1 for small signal gain and IFIELD = 1 for numerical gain.

For small signal gain only, the gain is computed first at the nozzle exit plane and then computed along the flow direction by integrating the rate equations in subroutine KINET.

The particular initial thermodynamic conditions, rotational J values (P or R branch), and initial vibrational temperatures are brought in through common/CAV2/. Then, for a particular vibration-rotation transition, the gain coefficient is given by:

$$g_{v'j'}^{vj} = \frac{8\pi^3}{3h} \left(\frac{M}{2\pi KT} \right)^{1/2} S_j F_j \left| R_{v',v'} \right|^2 \left[\frac{n_{vj}}{g_{vj}} - \frac{n_{v'j'}}{g_{v'j'}} \right] \quad (104)$$

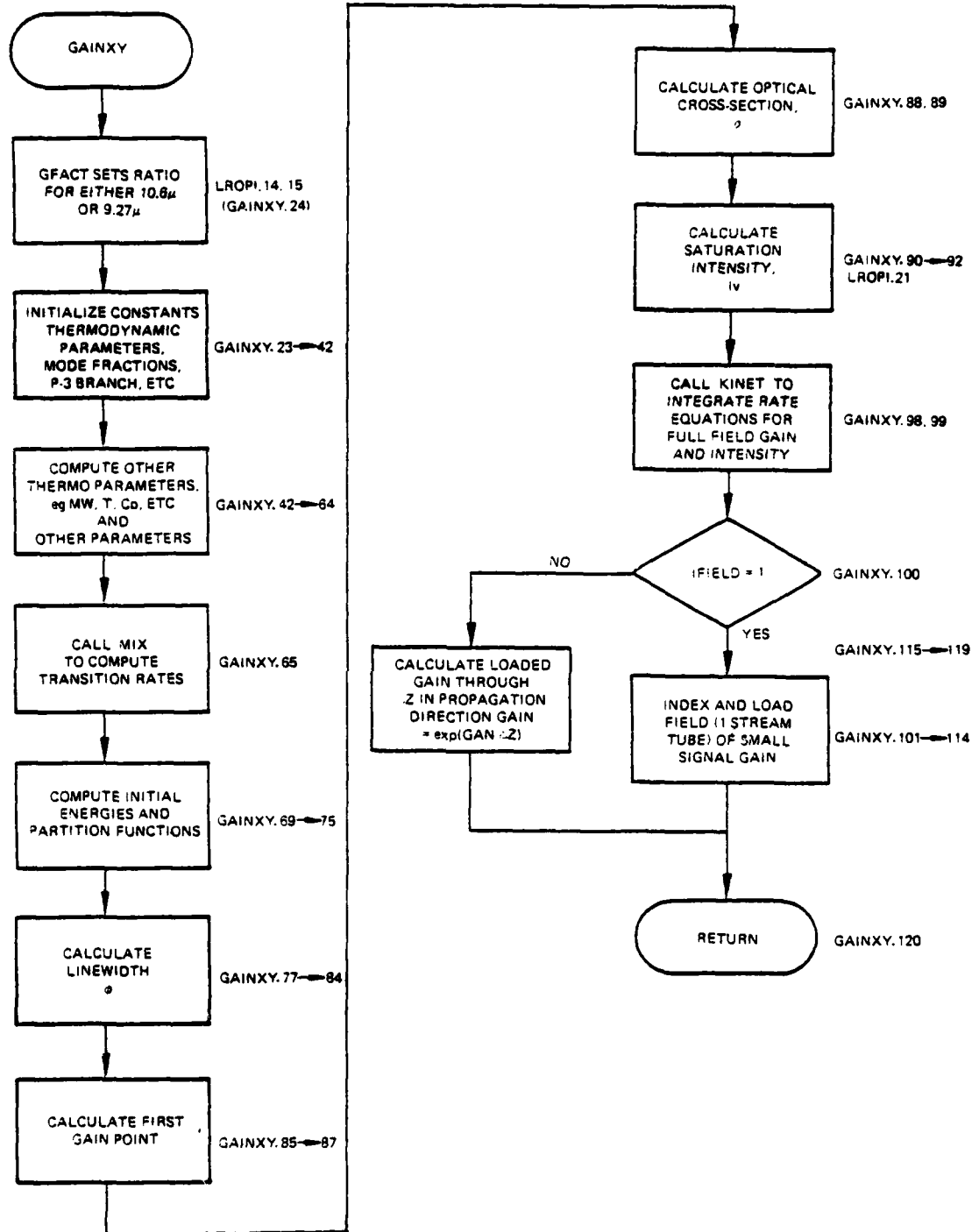


Figure 29. Subroutine GAINXY flow chart.

Where,

$$h = \text{Planck's constant} = 6.625 \times 10^{-27} \text{ erg}$$

$$M = \text{Mass of CO}_2 \text{ molecule} = 44 \times 1.66 \times 10^{-24} \text{ g}$$

$$K = \text{Boltzmann's constant} = 1.38 \times 10^{-16} \text{ erg/K}$$

$$S_J = \begin{cases} J + 1 & \text{for } J' = J + 1 \text{ (P-branch)} \\ J & \text{for } J' = J - 1 \text{ (R-branch)} \end{cases}$$

$$F_J = 1 + D_{v,v',m} \text{ where } M = \begin{cases} -(J+1) & \text{P-branch} \\ J & \text{R-branch} \end{cases}$$

$R_{vv'}$ = Vibrational matrix element for transition

ϕ = lineshape factor

$$= e^{-\xi^2} \text{erfc}(\xi)$$

$$= (\ln 2)^{\frac{1}{2}} \frac{\alpha_p}{\alpha_d}, \quad \begin{aligned} \alpha_p &= \text{pressure-broadened half-width} \\ \alpha_d &= \text{Doppler-broadened half-width} \end{aligned}$$

$$\alpha_p = \frac{n}{2\pi c} \sum_{\text{SPECIES}} x_i \bar{v}_{i-\text{CO}_2} \bar{\sigma}_{i-\text{CO}_2}$$

$$\alpha_d = \frac{v_o}{c} \left(\frac{2KT \ln 2}{M} \right)^{\frac{1}{2}}$$

n = total gas number density

c = speed of light = 3×10^{10} cm/s

x_i = mole fraction of the i th species

$\bar{v}_{i-\text{CO}_2}$ = mean velocity between CO_2 and i th species

$M_{i-\text{CO}_2}$ = reduced mass of $i-\text{CO}_2$ pair

$\alpha_{i-\text{CO}_2}$ = optical broadening cross-section

v_o = frequency of transition $(v, j) - (v', j')$

$$N_{VJ} = N_V f_J = N_V \frac{2J+1}{Q_{rot}^{(v)}} e^{-\frac{J(J+1)}{KT}} Q_{rot}^{(v)} \quad (105)$$

where,

$$Q_{rot}^{(v)} = \frac{T}{2\Theta_{rot}^{(v)}}$$

$$\frac{N_{VJ}}{g_{VJ}} = \frac{N_V}{g_V} \frac{\exp\left(-\frac{J(J+1)}{KT}\right)}{Q_{rot}^{(v)}}$$

$$\frac{N_V}{g_V} = N_{000} \exp(-\Theta_V/T_V)$$

Θ_V = Characteristic temperature of state

T_V = Vibrational temperature of state

The saturation intensity is calculated:

$$I_{SAT} = \frac{h\nu\beta}{\sigma} \quad (106)$$

where,

$h\nu$ = photon energy

β = lower laser level relaxation rate

σ = optical cross-section of the transition

Where R_{c2} is the EOVO transition rate $\sim (1/s)$, all the initial energies of the vibration levels are computed before entering subroutine KINET.

$$EOOVI = \frac{X_{CO_2} * 2349}{\epsilon \frac{hc * 2349}{KT_2} - 1}$$

$$EVOVI = \frac{X_{CO_2} * 2349}{\epsilon \frac{hc * 667}{KT_2}} - 1$$

$$EVOOI = \frac{X_{CO_2} * 2349}{\epsilon \frac{hc * 1388}{KT_1}} - 1$$

$$EN2I = \frac{X_{N_2} * 2331}{\frac{hc * 2331}{KT_{N_2}}} - 1$$

(107)

Where X_{CO_2} and X_{N_2} are mole fractions of CO_2 and N_2 , and T_1 , T_2 , T_{N_2} are vibrational temperatures. These vibrational temperatures and levels are shown schematically in Figure 30.

Gain is computed as a function of x by calling "KINET."

When the loaded numerical gain option is triggered (IFIELD) \neq 1), the full field (in X and Y) gain is calculated in KINET as a function of previous intensities and the field is updated when returned to GAINXY by propagating each local intensity through a ΔZ , with local gain GAN(I). The gain is thus recomputed for each point $G(J) = e^{G(J) \cdot \Delta Z}$.

Argument List

XIC	intensity array of propagation field
GAN	gain array of propagation field
NCV	cavity indicator
IFIELD	trigger for small signal gain (= 1) for full field loaded gain (\neq 1)

Commons Modified

/START/

TSI	static temperature (K)
PSI	static pressure (atm)
VI	gas velocity (cm/s)

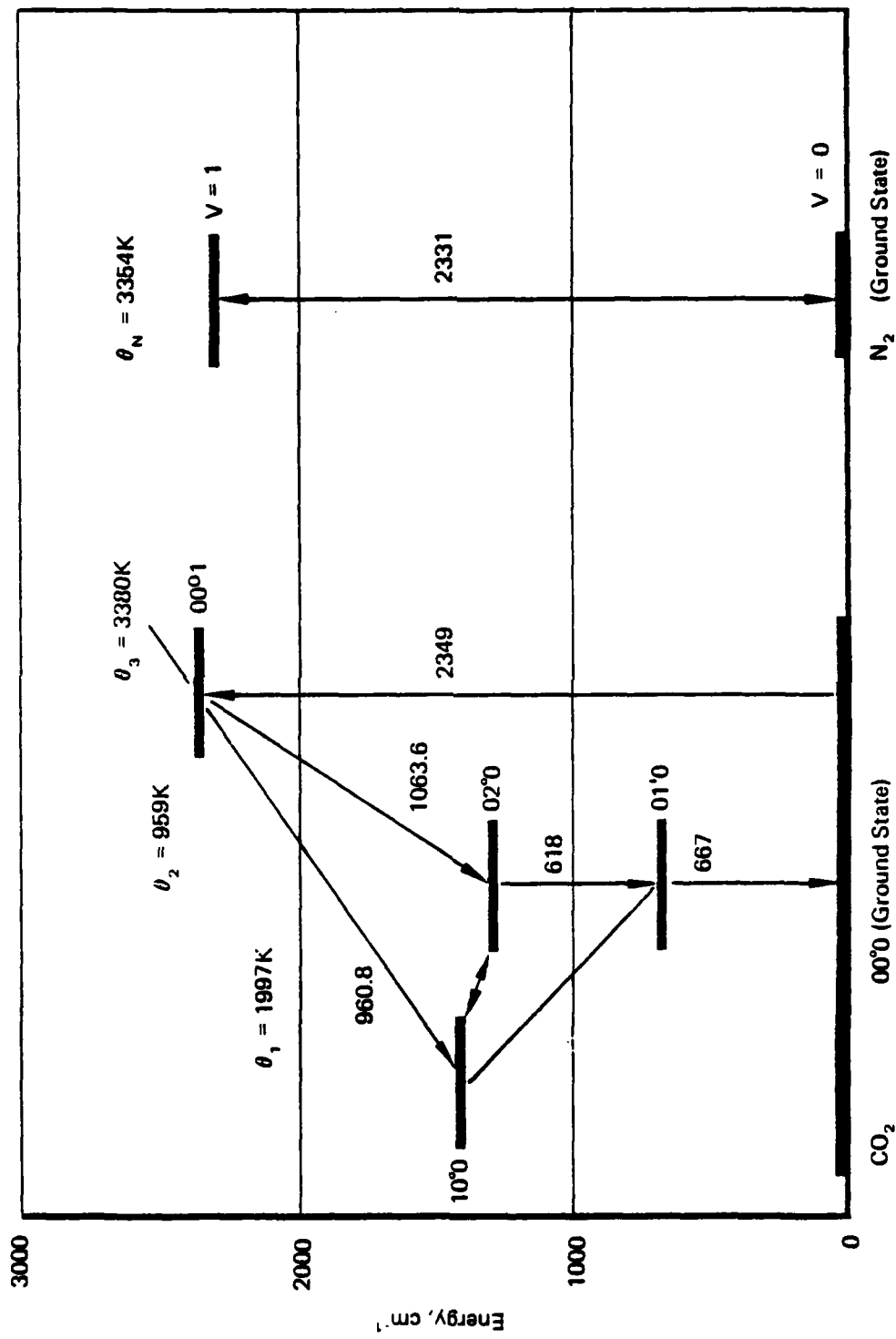


Figure 30. Characteristic temperature and energy levels.

E00VI Initial Energy (OOV level)
E0VOI Initial Energy (OVO level)
EN2I Initial Energy N₂ vibrational level
GAINI INITIAL GAIN

/PROPT/

TS static temperature (K)
PS static pressure (atm)
V gas velocity (cm/s)
RHO gas density (g/cm²)
RHON number density (cm⁻³)
CP specific heat @ constant pressure
GAMMA ratio of specific heats
R gas constant of mixture
B (ln 2) (3.78 x 10⁶)
XLAMB wavelength (λ)
HNU energy of photon of wavelength XLAMB
CPRM parameter to get Doppler broadened line width ratio

/MOLES/

XN2 mole fraction (N₂)
XCO2 mole fraction (CO₂)
XH2O mole fraction (H₂O)
XCO mole fraction (CO)
XO2 mole fraction (O₂)

/RATE/

RSTIM stimulated transition rate (s⁻¹)

/FACTOR/

MW molecular weight of gas mixture

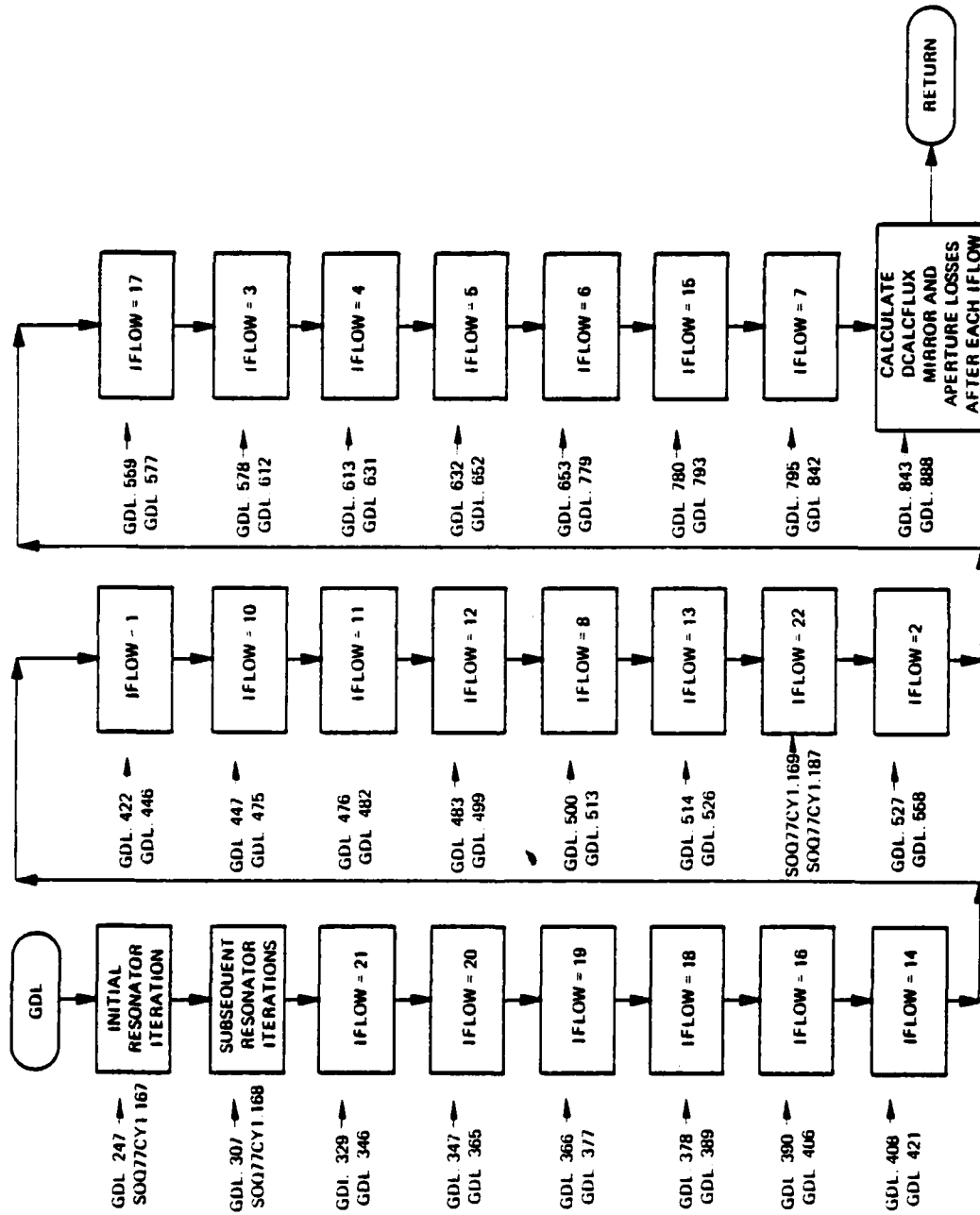


Figure 31. Subroutine GDL organization.

AG Avogadro's number
 GCON gain correction factor
 ROTUP upper rotational level (K)
 ROTLO lower rotational level (K)
 RCORR correction factor for optical x-section
 C speed of light (cm/s)

SUBROUTINE GAINXY 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE GAINXY(XIC,GAN,NCV,IFIELD)
NUMERICAL GAIN ROUTINE
THIS ROUTINE CALCULATES: 1. SMALL SIGNAL GAIN FOR USE IN SIMPGG
                        2. NUMERICAL LOADED GAIN
*****
IFIELD = 1 FOR SMALL SIGNAL GAIN ONLY
*****
LEVEL 2: XIC,GAN,XC
COMMON/STANT/TSI,PSI,VI,EUVVI,EUVVI,EUVVI,ENZI,GAINI
COMMON /GFACR/ GFAC(2)
COMMON/PHOPT/TS,PS,V,MMU,MMUN,CP,GAMMA,H,B,XLAMB,MNU,CPRM
COMMON/MULES/XN2,ACU2,XM2U,XLU,AU2
COMMON/ENERG/EN2,EUVV,EUVU,EVUU
COMMON/HATE/HN2,HC3,HC2,HPUMP, NSIIM
COMMON/FACTER/AM,AG,GCON,MUTUP,MUTLU,NCUMH,C
COMMON/CAV2/AC(5),YC(5),ZC(5),NX(5),NY(5),NS(5),XMC(5),YMC(5),
2 NGTYP(20), SSGAIN(190,5),SATIN(5),BETA(5),RHOS(5),
3 VEL(5),GAM(5),XMAC(5),TV1(5),TV2(5),TV3(5),TVN2(5),TSCAV(5),
4 PSCAV(5),PBCH(5),FN2(5),FCU2(5),FM2U(5),FCU(5),FO2(5),
5 TITLE(20),AVG(5),NSYM
DIMENSION XIC( 1 ),GAN( 1 )
CALL CPUTIM(IISRT)
TSI=TSCAV(NCV)
WLFAC=1.
IF (GFAC.NE.1.) WLFAC = 10.0/9.27
*****
SLUPIN = 2.15
*****
PSI=PSCAV(NCV)
VI=VEL(NCV)
PB=PBCH(NCV)
XN2=FN2(NCV)
ACU2=FCU2(NCV)
XM2U=FM2U(NCV)
XCU=FCU(NCV)
XU2=FU2(NCV)
T1 =TV1(NCV)
T2 =TV2(NCV)
T3 =TV3(NCV)
TN2=TVN2(NCV)
TS = TSI
PS = PSI
V = VI
HO = 8.317E7
C GFAC MODIFIES GAIN
GCON = .991E-14*PB*GFAC(NCV)
MUTUP = (PB-1.)*PB*.556
MUTLU = PB*(PB+1.)*.561
AG = 6.023E23
XN2 = XN2*ACU
GAINXY 2
GAINXY 3
GAINXY 4
GAINXY 5
GAINXY 6
GAINXY 7
GAINXY 8
GAINXY 9
GAINXY 10
LNUP1 13
GAINXY 11
GAINXY 12
GAINXY 13
GAINXY 14
GAINXY 15
GAINXY 16
GAINXY 17
GAINXY 18
GAINXY 19
GAINXY 20
GAINXY 21
GAINXY 22
GAINXY 23
LNUP1 14
LNUP1 15
GAINXY 24
GAINXY 25
GAINXY 26
GAINXY 27
GAINXY 28
GAINXY 29
GAINXY 30
GAINXY 31
GAINXY 32
GAINXY 33
GAINXY 34
GAINXY 35
GAINXY 36
GAINXY 37
GAINXY 38
GAINXY 39
GAINXY 40
GAINXY 41
GAINXY 42
LNUP1 16
LNUP1 17
GAINXY 44
GAINXY 45
GAINXY 46
GAINXY 47

```

```

XNW = 28.016*XN2+44.011*XC02+18.016*XM20+32.0*XC02
XU2FAC=20.939
IF (GFACI.NE.1.) XU2FAC=10.520
HCUHM = .604* XN2 /SQR(17.12/22.005) + XC02 + .292*XM20/
X SQR(12.783/22.005)+.046*XC02/SQR(XU2FAC/22.005)
SIGMA = 13.E-15
HCUHM = SIGMA*HCUHM*.105E-1
C = 3.E10
B = .09315*0.03/0EB
H = 0.025E-27
XLAMB = 1.434/(1380.*(PH-1.)*PB+.556*PB*(PB+1.)*.561)
HNU = H*C/XLAMB
CPHM = HCUHM*C*XLAMB/SQR(B)
N = H0/XNW
GAMMA = (7.*(XN2+XC02+XU2)+8.*XM20)/(5.*(XN2+XC02+XU2)+6.*XM20)
HMO = P5/N/TS*1.013E6
GAM(NCV)=GAMMA
XMACH(NCV)=V1/SQR(GAMMA*N*151)
HMUS(NCV)=HMO
CALL MIA
BETA(NCV)=HC2
HMON = HMO/XNW*AG
CP = 3.5*H0 *(XN2+XC02+XU2+8./7.*XM20)
EU0V1 = XC02*2349./(EXP(1.434*2349./13)-1.)
EU0V1 = XC02*1334./(EXP(1.434*667./T2)-1.)
EU0V1 = XC02*1388./(EXP(1.434*1388./T1)-1.)
EN2I = XN2*2331./(EXP(1.434*2331./TN2)-1.)
Q1 = 1./(1.-EXP(-1997./T1))
Q2 = 1./(1.-EXP(-960./T2))**2
Q3 = 1./(1.-EXP(-3380./T3))
X000 = XC02/(Q1*Q2*Q3)
C CALCULATE LINEWIDTH
APAO = CPHM*HMON
WUHM = .8326*APAO
IF(WUHM.GT.10.) GO TO 40
PHI = EXP(WUHM**2)*ENFC(WUHM)
GO TO 41
40 PHI = 0.67764/APAO
41 CONTINUE
TFAC = TS*(1-1.5)
GAIN = GCUN*TFAC*HMON*X000*PHI*(.556*EXP(-3380./T3)-HUTUP/TS)
X -.561*EXP(-1997./T1)-HUTU/TS)
C OPTICAL CROSS SECTION
BIGSIG = GCUN*TFAC*PHI*EXP(-HUTUP/TS)*.556
C SATTIATION INTENSITY
SATIN(NCV)=HNU*HC2/BIGSIG/1.E7
IF (NGTYP(NCV).EQ.0) SATIN(NCV)=SATIN(NCV) * SLUPIN
SATIN(NCV) = SATIN(NCV) * WFACT
NSTIM = 0.0
GAINI = GAIN
IAXMAX=NX(NCV)
IY=NY(NCV)/(NSYM+1)
DACAV=XC(NCV)/IAXMAX
C CALCULATE GAIN AS A FUNCTION OF X
CALL KINET(XIC,GAN,IAXMAX,DACAV,IFIELD,IY)
IF(IFIELD.NE.1) GO TO 980
C INITIALIZE SMALL SIGNAL GAIN
DO 300 I = 1,IAXMAX
300 SSGAIN(I,NCV)=GAN(I)
SATINK=SATIN(NCV)/1000.
WRITE(6,100) GAN(NCV),XMACH(NCV),HMUS(NCV),BETA(NCV),SATINK
100 FORMATT(2)MUNRESULTS FROM KINETICS DECK/1X,8H6GAMMA = ,G12.5,4X,15HMA
XCH NUMBER = ,G12.5,4X,10HUNENSITY = ,G12.5,4X,7HBETA = ,G12.5,
X 4X,8HSATIN = ,G12.5//27X,4(18H XNEP GO(XNEP))
GAINXY 108
DO 101 I=1,IAXMAX
GAN(I,IAXMAX)=100.*SSGAIN(I,NCV)
GAINXY 109
101 GAN(I)=(2*I-1)*DACAV/2.
WRITE(6,102) (GAN(I),I=1,IAXMAX) ,I=1,IAXMAX
GAINXY 110
GO TO 982
GAINXY 111
102 FORMATT(6)Z5X,8FY,3//)
GAINXY 112
GAINXY 113
GAINXY 114

```

C	CALCULATE LOADED GAIN	GAINAY	115
980	DELTAZ=ZC(NCV)/NS(NCV)/2.	GAINAY	116
	MUT = I*MAX*Y	GAINAY	117
	DO 981 J=1,MUT	GAINAY	118
981	GAN(J)=EXP(GAN(J)*DELTAZ)	GAINAY	119
982	RETURN	GAINAY	120
	END	GAINAY	121

13. SUBROUTINE GDL

a. Purpose -- Subroutine GDL is the main driver program for resonator and optical train calculations. It is here that the information about each resonator element is stored, as well as the order in which they are applied to the beam. Figure 31 shows the Subroutine GDL organization.

b. Formalism -- Subroutine GDL controls the iterative procedure of starting with a given field established in the main program (SOQ) and propagates this field through the resonator. Eventually, the mode which loses the least power (in the case of a bare resonator) or gains the most power (in the case of a loaded resonator) will predominate since the other modes will be suppressed due to relative power loss. For the degenerate case when two or more modes are competing for the status of lowest loss mode, the field will usually fail to converge to a single mode shape, since there is no unique mode for that eigenvalue.

c. Fortran -- To accomplish the above, GDL contains several fundamental arrays. One is the singly dimensioned CU array in which the field is stored. For a given point (x(I), x(J)) the field value is stored in the complex location.

CU (I + (J-1) * NPTS)

Common /MELT/ contains CU as well as the work array CFIL, the coordinate array x, the location of the optical axis (DRX and DRY), and the iteration number NITER. This common is shared by most of the routines in the deck. The other major arrays are the ABC array, the IGDL array, and the GNOT array. During the first iteration of a particular run, GDL reads input from unit IN in the form of namelists and titles. The order of resonator elements to be met by the beam is controlled by the order in which the \$CONTRL cards are read. These contain the IFLOW parameters which designate specific elements, as follows:

NAMELIST/CONTROL/IFLOW, SNOTE, IPLOTS

IFLOW CONTROLS THE FLOW OF CALCULATIONS THROUGH GDL

- = 1 CAVITY ELEMENT, READS CAVTY1, CAVTY2.
(from CAVITY)
- = 2 MIRROR ELEMENT, READS MIRROR
- = 3 VAMP ELEMENT, READS PROPGT
- = 4 APERTURE ELEMENT, READS APTUR
- = 5 THERMAL BLOOMING, READS BLOOM
- = 6 INTERPOLATE FIELD OVER SMALLER AREA, READS CUTOUT
- = 7 TEST FOR CONVERGENCE OF ITERATION, NO INPUT
- = 8 PLOT FIELD DISTRIBUTION, READS TITLE
- = 9 RETURN CONTROL TO CALLING PROGRAM, NO INPUT
- = 10 READ AND/OR WRITE CU ON DISK, READS DISKIT
- = 11 AERO WINDOW R.M.S. PHASE MODEL, NO INPUT
- = 12 SCALING ROUTINE . . .MULTIPLIES ENTIRE FIELD,
READS MULT
- = 13 FLIPS THE FIELD ABOUT THE y-AXIS, NO INPUT
- = 14 SINUSOIDAL DENSITY VARIATIONS, READS SINDEN
- = 15 REGRIDS FIELD TO LARGER SIZE, READS REGRID
- = 16 CU PUNCHED ON CARDS, NO INPUT
- = 17 MIRROR THERMAL BL MODEL, READS THRML
- = 18 SPIDER ROUTINE, READS SPIDR
- = 19 AXION ROUTINE, READS AXICON
- = 20 PROPAGATE IN R-THETA SPACE, READS RPROP
- = 21 REMOVES OR ADDS BACK BEAM CENTER, READS CENTER
- = 22 FLIPS THE BEAM ABOUT THE x-AXIS, NO INPUT

IPLOTS is the printer plot selector. IPLOTS=ABCDE, where A=1 selects R-theta plots, B=1 selects iso-intensity plot, C=1 selects x-axis plot, D=1 selects diagonal plot, and E=1 selects y-axis plot: example, IPLOTS = 1001 selects

iso-intensity and y-axis plots in x-y coordinates. The order of IFLOW numbers for a given resonator is then stored in the IGDL array for future iterations. In the same manner the associated titles are stored in the GNOT array.

Usually for a given IFLOW there is another associated namelist containing relevant element parameters. Once read in, these numbers are stored in ABC (I,J,K) where I indicates the parameter for the J the element of type K. The number (J) of the element is stored in common ZIP, which is equivalenced to the ICAVZ array. At the beginning of each iteration most of ICAVZ is filled with zeros so that the center index of the ABC array is correctly identified. At the end of each iteration, the current field is compared with that of the previous iteration in two ways: (1) the cutout and interpolated feedback field is compared and (2) the full field just before the hole-coupling mirror is compared. When the differences between two consecutive iterations fall within given tolerances (10% for the feedback field, 2% for the hole-coupler field and 0.7% for the power at the output of the resonator), the field is said to have converged, i.e., the lowest loss mode has been selected. A more detailed description of the meaning of each IFLOW, its function, and its associated namelist, if any, follows:

IFLOW = 1 (GDL. 422→GDL.446)

A GDL cavity is applied to the field. NEWCAV is calculated to see if the beam has been in the cavity before. The namelist used in CAVTY1.

CALLS CAVITY.

NAMelist/CAVTY1/NCAVNO, ILR, NSTE, NPLT, ZPROP1, ZPROPO

NCAVNO IS THE NUMBER ASSIGNED TO CAVITY FOR IDENTIFICATION

ILR INDICATES DIRECTION OF FIELD THROUGH CAVITY

= -1 RIGHT TO LEFT

= +1 LEFT TO RIGHT

NSTE CONTROLS TYPE OF VAMP CODE BETWEEN SEGMENTS

= 1 CONSTANT MESH WITH SETUP

= 2 VARIABLE MESH WITH SETUP (EXITS VAMP AT END OF ELEMENT)

= 3 VARIABLE MESH WITH SETUP (REMAINS IN VAMP)

- = 4 USE EXISTING PROPAGATING MATRIX (EXITS VAMP)
- = 5 USE EXISTING PROPAGATING MATRIX (REMAINS IN VAMP)

NPLT CONTROLS INTERMEDIATE PRINTOUT FOR CAVITY

- = 0 NO PRINTOUT
- = 1 PRINT FIELD BEFORE AND AFTER GAIN, AND GAIN COEFFICIENT

ZPROPI IS PROPAGATION DISTANCE FROM PREVIOUS OPTICAL ELEMENT TO CAVITY.

ZPROPO IS PROPAGATION DISTANCE FROM CAVITY TO NEXT OPTICAL ELEMENT

IFLOW = 2 (GDL.527→GDL.558)

Here the parameters necessary for application of a mirror are set up.
The namelist read is MIRROR. CALLS MIRROR

NAMELIST/MIRROR/ANGXX, ANGY, RAD, DIAOUT, DIAIN, XMPOS, YMPOS, RMIR,
X DELIA, DISTF, DDUTY, DINY, RANULS, PHIAT

ANGXX IS TILT IN x-DIRECTION - RADIANS (WRT OPT. AXIS)

ANGY IS TILT IN y-DIRECTION - RADIANS (WRT OPT. AXIS)

RAD IS RADIUS OF CURVATURE OF SPHERICAL MIRROR

DIAOUT IS OUTSIDE DIAMETER OF MIRROR

DIAIN IS INSIDE DIAMETER OF MIRROR

XMPOS IS X-DISPLACEMENT OF MIRROR FROM OPTICAL AXIS

YMPOS IS Y-DISPLACEMENT OF MIRROR FROM OPTICAL AXIS

RMIR IS REFLECTIVITY OF MIRROR

DELTA IS CENTER-TO-EDGE DISTORTION FACTOR (CM)

DISTF IS MIRROR DISTORTION FACTOR (DEFLECTION=DISTF*I*
(1.0-RMIR))

RANULS IS OUTSIDE RADIUS OF ANNULAR BEAM (IF APPLICABLE)

DDUTY FLAGS THE TYPE OF APERTURE APPLIED -

.EQ. 0 - CIRCULAR APERTURE DEFINED AS ABOVE

.NE. 0 - RECTANGULAR APERTURE, DIAOUT HIGH (X) BY
DDUTY WIDE (Y)

DINY IS SIMILAR TO DDUTY FOR INSIDE DIMENSIONS

PHIAST IS THE ANGLE OF INCIDENCE OF THE BEAM IN DEGREES

IFLOW = 3 (GDL.578→GDL.612)

For this IFLOW, a propagation step is applied. Relevant parameters are found in namelist PROPGT. CALLS STEP.

NAMelist/PROPGT/DELZ, RDCURV, WINDOX, WINDOK, IIFG, IITR, IIPS

DELZ IS PROPAGATION DISTANCE

RDCURV IS RADIUS OF CURVATURE OF PHASE FRONT

IF (ABS (RDCURV) .LT.0.5) USE RADCUR OF PREVIOUS

MIRROR

WINDOX IS X-SPACE DATA WINDOW FOR FFT

WINDOK IS K-SPACE DATA WINDOW FOR FFT

IIFG IS A VAMP CONTROL PARAMETER

= 1 FOR CONSTANT MESH

= 2 FOR VARIABLE MESH

IITR IS ANOTHER VAMP CONTROL PARAMETER

= 0 NO INVERSE TRANSFORM

= 1 INVERSE TRANSFORM BACK TO REAL SPACE

IIPS IS FOR CORRECTION OF PLANE AND SPHERICAL PHASE

FRONTS

= 0 NO CORRECTION

= 1 PLANAR CORRECTION ONLY

= 2 QUADRATIC CORRECTION ONLY (NOT OPERATIONAL)

= 3 BOTH

IFLOW = 4 (GDL.613→GDL.631)

Here an aperture is applied. IF DOUT and DIN are both less than 0, SLIVER is called. If both are greater than or equal to zero, APRTR is called. The relevant namelist is APTUR.

AD-A103 285

UNITED TECHNOLOGIES CORP WEST PALM BEACH FLA
SYSTEM OPTICAL QUALITY USERS GUIDE. PART 2.(U)
MAR 80 J L FORGHAM, S S TOWNSEND

F/6 20/5

F29601-77-C-0025

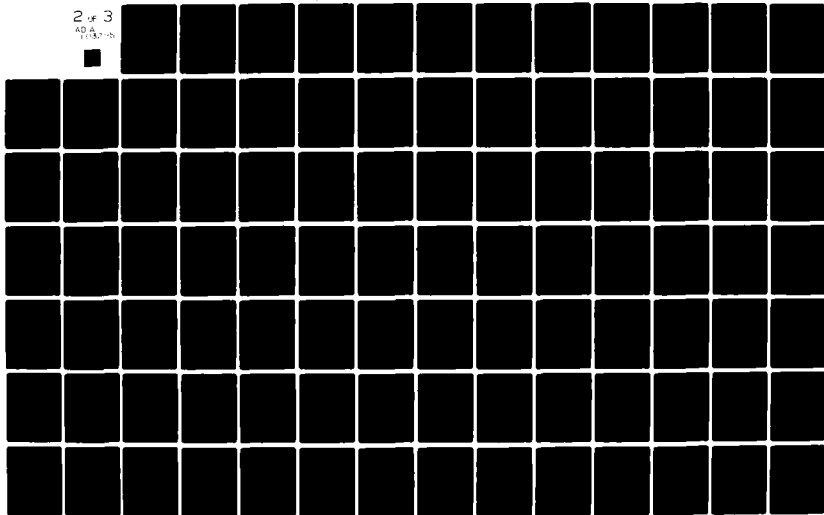
UNCLASSIFIED

AFWL-TR-79-141-PT-2

NL

2 of 3

AD-A
13-5



NAMelist/APTUR/DOUT, DIN, XPOS, YPOS, YOUT, YIN

DOUT IS OUTSIDE DIAMETER OF APERTURE

DIN IS INSIDE DIAMETER OF APERTURE

XPOS IS x-DISPLACEMENT OF APERTURE FROM OPTICAL AXIS

YPOS IS y-DISPLACEMENT OF APERTURE FROM OPTICAL AXIS

YOUT FLAGS THE TYPE OF APERTURE APPLIED -

.EQ.0 - CIRCULAR APERTURE DEFINED AS ABOVE

.NE.0 - RECTANGULAR APERTURE, DOUT HIGH (X) BY
YOUT WIDE (Y)

YIN IS SIMILAR TO YOUT FOR INSIDE DIMENSIONS

IFLOW = 5 (GDL.632-GDL.652)

Thermal Blooming is applied to the complex field. BLOOM is read in and subroutine TBLCOM is called.

NAMelist/BLOOM/ALFA, SCP, T, RHO, ZLEN, NSTEPS, INPT, NPROP, AXIAL, DT

AFLA = MEDIUM ABSORPTION COEFFICIENT, CM^{-1}

SCP = MEDIUM SPECIFIC HEAT, J/GM-DEG K

T = MEDIUM TEMPERATURE, DEG K

RHO = MEDIUM DENSITY, GM/CM³ (OR TRANSVERSE VEL.
IF .GT.1.)

ZLEN = MEDIUM THICKNESS ALONG OPTICAL AXIS

NPROP = PROPAGATION PARAMETER. . .SAME AS NSTE IN
CAVITY

NSTEPS = NUMBER OF ELEMENTS IN SUBSYSTEM, .GE. 1

INPT = .NE.0 FOR INTERMEDIATE FIELD PLOTS

AXIAL = AXIAL VELOCITY (CM/SEC) IF .GT. 0, USES
AXIAL BLOOMING

DT = BEAM ON TIME FOR THERMAL BOUNDARY LAYER
GROWTH IN TRANSIENT BLOOMING CALCS. IF
DT.GT.0 USES TRANSIENT BLOOMING

IFLOW = 6 (GDL.653-GDL.779)

For this option the field can be cut out and interpolated from one region size to another. The number of points is not changed. If CUSMF is not equal to zero, the field-averaged feedback field is stored on unit 8 and the convergence checks are made on the feedback field and the pre-HCM field which is stored on unit 7 temporarily. The field for the bare-resonator is renormalized at this point to unit maximum intensity. Namelist CUTOUT has the information for the new region in it as well as other parameters.

NAMelist/CUTOUT/DIBEAM, OVLAP, DXR, DYR, MAXIT, AVCUSM, CUSMF

CUSMF = 1. FOR NORMAL LOADED RESONATOR CUTOUT

CUSMF = 0. AVOIDS WRITING FIELD ON 8 AND AVOIDS NORMALIZING FIELD, CHANGES TO THE NEW COORDINATES, THEN RETURNS.

DIBEAM IS THE DIAMETER OF BEAM FOR NEXT ITERATION

OVLAP IS $DCALC = OVLAP * DIBEAM$

DXR IS POSITION OF ITERATIVE BEAM REL. TO OPTICAL AXIS

DYR IS THE SAME

MAXIT IS THE MAXIMUM NUMBER OF ITERATIONS

AVCUSM AVERAGES PREVIOUS AND NEXT ITERATION GUESS IN THE

HOPE OF RAPID CONVERGENCE...=0 NO AVE, = .5 HALF AND HALF

IFLOW = 7 (GDL.795→GDL.842)

There is no namelist associated with this option. The convergence check on the power is made here. If the solution has not yet converged, the gain/phase information is updated by a call to REGAIN, then the resonator is restarted for the next pass.

IFLOW = 8 (GDL.500→GDL.513)

If the parameter plot is non-zero in namelist START in SOQ, this IFLOW will generate printer plots by a call to IPLOT. Namelist PLOT is read.

NAMelist/PLOT/TITLE RADPLT

TITLE IDENTIFIES THE POSITION OF EACH STATION PLOTTED

RADPLT CONTROLS THE TYPE OF PLOT

= 0.0 FOR X,Y PLOTTING (X-AXIS, Y-AXIS, DIAGONAL)

= 1.0 FOR RADIAL PLOTTING AT VARIOUS THETAS

IFLOW = 9

This IFLOW only results in the return to the main program, SOQ.

IFLOW = 10 (GDL.447→GDL.475)

This option allows the field to be read in from or read to a specific unit in standard SOQ format. It calls no peripheral subroutines and reads the unit designation from namelist DISKIT.

NAMelist/DISKIT/IREAD, IWRITE, IORD, IADD

IREAD IS THE DISK # TO BE READ OFF/ON... IF=0...DON'T

READ

IWRITE IS THE DISK # TO BE WRITTEN ON... =0...DON'T WRITE

IORD IS THE ORDER = 1, READ FIRST
=-1, WRITE FIRST

IADD = 1 UPDATES IWRITE BY 1 FOR SUCCESSIVE ITERATIONS

IFLOW = 11 (GDL.476→GDL.482)

This option applies an aerodynamic window to the complex field. It reads no namelist and calls AEROW to perform the calculation.

IFLOW = 12 (GDL.485→GDL.499)

The field can be scaled using this option. At the same time the x array can also be magnified. No subroutines are called and MULT is read.

NAMelist/MULT/TRANS, XMAG

TRANS IS TRANSMISSION OF ELEMENT

XMAG IS MAGNIFICATION FACTOR FOR THE X-ARRAY

IFLOW = 13 (GDL.514→GDL.526)

This option flips the field about its y-axis. No namelists are read and no subroutine called.

IFLOW = 14 (GDL.408→GDL.421)

This option imposes a sinusoidal density (phase) variation to the existing complex field. It calls no subroutines, but it reads SINDEN for information on the sine wave.

NAMelist/SINDEN/NBEAM, AWL

NBEAM IS THE NUMBER OF CYCLES PER X-CALCULATED REGION

AWL IS THE AMP/WL OF THE SINUSOIDAL VARIATIONS

IFLOW = 15 (GDL.780-GDL.793)

The field can have superimposed on it a different number of mesh points. The spacing between two adjacent points does not change unless RGRD is called. Just the number of points in the mesh changes. If the number of points is increased, RGRD adds zeros to the outside of the existing region. This option reads namelist REGRID

NAMelist/REGRID/NGRD

NGRD IS NO. OF FIELD POINTS ACROSS REGRIDDED DCAL

IFLOW = 16 (GDL.390-GDL.406)

In this IFLOW, no subroutine is called and no input is read. The field and coordinates are written format to TAPE 4 in cards to be punched.

IFLOW = 17 (GDL.559-GDL.557)

Quiescent thermal gradients are imposed by this option. Namelist THRML is read and subroutine THERML is called.

NAMelist/THRML/ALPHAM, CONMIR, ALPHAG, RHOGAS, TAU, TIN, REFMIR, CONGAS

THRML IS THE NAMelist FOR BOUNDARY LAYER THERMAL LENS

CALCULATIONS

ALPHAM = MIRROR DIFFUSIVITY (CM²/SEC)

CONMIR = MIRROR THERMAL CONDUCTIVITY (WATTS/CM-SEC)

ALPHAG = THERMAL DIFFUSIVITY OF GAS HEATED BY MIRROR
(CM²/SEC)

CONGAS = THERMAL CONDUCTIVITY OF GAS HEATED BY MIRROR
(WATT/CM-SEC)

RHOGAS = DENSITY OF GAS HEATED BY MIRROR (GM/CC)

TAU = BEAM ON TIME FOR BOUNDARY LAYER GROWTH (SEC)

TIN = INITIAL TEMPERATURE OF GAS & MIRROR (DEG K)

REFMIR = MIRROR REFLECTIVITY (OBTAINED FROM MIRROR
INPUT)

THERMAL MAY BE APPLIED AFTER ANY MIRROR TO ALTER THE GAIN-
PHASE DUE TO HEATING OF THE QUIESCENT BOUNDARY LAYER
ADJACENT TO THE MIRROR SURFACE.

IFLOW = 18 (GDL.378→GDL.389)

With IFLOW = 18, a spider obscuration can be applied. Subroutine SPIDER
is called using the information read in with namelist SPIDR.

NAMELIST/SPIDR/NSPD, WIDTH, THETA, XSPC, YSPC, DIH

NSPD = NUMBER OF STRUTS IN SPIDER (MAX=6)

WIDTH = WIDTH OF SPOKES IN SPIDER

THETA = ANGLE OF INDIVIDUAL SPOKES OF SPIDER

XSPC = x-LOCATION OF CENTER OF SPIDER

YSPC = y-LOCATION OF CENTER OF SPIDER

DIH = HUB DIAMETER

IFLOW = 19 (GDL.366→GDL.377)

This option allows for the application of an axicon. Subroutine AXICV
is called after namelist AXICON is read.

NAMELIST/AXICON/CAPR, EXPAND, ROC, DISP, TILT

CAPR IS THE OUTSIDE RADIUS OF THE ANNULAR EXTRACTION BEAM.
(EXPAND.EQ. .TRUE.) MEANS THE BEAM IS GOING FROM CIRCULAR
TO ANNULAR IN CROSS-SECTION

ROC = RADIUS OF CURVATURE OF THE FIELD IN PHYSICAL
SPACE

DISP = DISPLACEMENT OF AXICON FROM CENTER ALONG
X-AXIS
TILT = ANGLE (RADIAN) OF AXICON TILT FROM DIRECTION
OF PROP.

IFLOW = 20 (GDL.347→GDL.365)

This option propagates an unrolled annulus. After reading in namelist RPROP, it then calls subroutines RSTEP to perform the propagation and POWR to determine the power after propagation.

NAMelist/RPROP/DELZR, DELZTH, WINDOX, WINDOK

DELZTH IS PROPAGATION DISTANCE FOR THE RADIAL COORDINATE
DELZTH IS PROPAGATION DISTANCE FOR THE ANGULAR COORDINATE
*** (DELZR .NE. DELZTH) MEANS YOU ARE MAKING AN
EQUIVALENT COLLIMATED BEAM PROPAGATION STEP
IN R-THETA COORDINATES***

WINDOX IS X-SPACE DATA WINDOW FOR FIT

WINDOK IS K-SPACE DATA WINDOW FOR FFT

IFLOW = 21 (GDL.329→GDL.346)

This option allows for the removal of the center of the beam which is then stored on unit 20, or it can allow for the addition of a field read from unit 20 modified by a phase change. This work is all done in subroutine FIELDS using the information read in from namelist CENTER

NAMelist/CENTER/DSM, REMOVE, PHIARB

DSM IS THE DIAMETER TO BE REMOVED AND LATER ADDED TO THE
MAIN BEAM

REMOVE FLAGS THE ACTION

.TRUE. IF THE CENTER PORTION OF THE BEAM IS TO BE REMOVED

.FALSE. IF THE REMOVED PORTION IS TO BE ADDED BACK TO THE
BEAM

PHIARB IS AN ARBITRARY PHASE CHANGE ADDED TO THE CENTRAL
PORTION

IFLOW = 22 (SQ77CY1.169→SQ77CY1.181)

This option flips the field about the x-axis. No input is required and no subroutines are called.

Argument List

IN - INPUT UNIT FOR RESONATOR DATA

RESTRT - NEW OR OLD RESONATOR?

ABC - PARAMETER ARRAY

NITER - CURRENT ITERATION

IB - INPUT UNIT # OF OLD FIELD

IFLAG == 1 TRANSFERS TO OLD ENTRY POINT - READS FIELD FROM IB/
CONTINUES.

ABC and NITER can be redefined by this subroutine.

Common Variables Modified:

The common variable not modified by GDL are:

WL, NPTS, NPY, RADCUR, WNOW and NREG. Note that NBC is modified by its equivalence with IGDL and IDIR.

SUBROUTINE GDL 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE GDL(IN,RESTRT,ABC,NITER,IB,IFLAG)          GDL      2
C OPTICAL CALCULATIONS ROUTINE ROUTINE              GDL      3
C BY MEANS OF THE INPUT (BCONTRL) THE USER INSTRUCTS THIS ROUTINE GDL      4
C TO DIRECT THE CALCULATION OF OPTICAL EFFECTS OF APERTURES. GDL      5
C MINIMUMS, CAVITIES, ETC.                          GDL      6
C IFLAG=1 TRANSFERS TO OLD AUTO ENTRY POINT          GDL      7
C.....C                                             GDL      8
C IN IS UNIT CONTAINING INPUT DATA FOR CONFIGURATION GDL      9
C RESTRT IS CONTROL FOR RESTARTING CALCULATIONS FROM PREVIOUS RUN GDL     10
C   = .TRUE. IF RESTARTING                          GDL     11
C   = .FALSE. IF NOT                                GDL     12
C.....C                                             GDL     13
C                                                     GDL     14
LEVEL 2: CU,CFIL2,CFPL                                GDL     15
COMMON/MELT/CU(1638),CFIL(16512),A(128),WL,NPTS,NPY,UNX,UNY GDL     16
COMMON/MHPHOP/HAUCUR,ANGX,ANGY                       GDL     17
COMMON/ WAY / #NUM,NNEG,HAPTH                         GDL     18
COMMON/ZIP/ICAV,IMIN,ISTEP,NUS,IAP,IPTT,ITHAN,ITHML,IAX,INSIP, GDL     19
```

X ICUT,MLT,DK,ITM,ICER,NCT	GUL	20
COMMON /QAZ/ APLT(30,20), NBC(180), SAVE(10)	GUL	21
CUMM /INITL/ INT	GUL	22
DIMENSION IDIR(4,24),IGOL(99),ABC(12,20,9),CFFL(16384),IUSK(4,9),	CIUFLA	1
XTAY(262),XK(128),XOOM(128),EMMOR(10),TITLE(20),CFIL2(16384),Y(128)	GUL	24
X,ZLI(12),ZLO(12),GNOTE(20),GNUT(50,20),THETA(6),CPR(4),XPNU(4),	CIUFLA	2
X DSMM(20),HMV(20),PHIA(20),NCURVE(4),DSP(4),TLT(4),ICAVZ(16)	GUL	26
DIMENSION IPLTS(50)	GUL	27
COMPLEX CFFL,CFIL2,CU,CFIL,CPLNT,CFACT,CUD	GUL	28
LOGICAL INIT,HESTH,WHY,EXPAND,XPNU,REMOVE,HMV	GUL	29
EQUIVALENCE (NBC(1),IGOL(1)),(NBC(100),IDIR(1,1))	CIUFLA	3
EQUIVALENCE (CFIL(1),CFFL(1)),(CFIL2(1),CU(1)),(ICAV,ICAVZ(1))	GUL	32
DATA IFLW,TITLE /9,20*4M / , IPLTS / 0 /	GUL	33
DATA NCAVNU,ILR,NSTE,NPLT,ZPHUPI,ZPHUPU /0,1,0,2*0./	GUL	34
DATA ANGX,ANGYY,RADC,UJAUUT,UJAIN,XPPOS,YMPOS,HMIR,UELTA,DISTF	GUL	35
X /0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 1,0, 0,0, 0,0/	GUL	36
DATA MANULS, DOUTY, UINY, PHIAST	CI0ASTG	2
X /0,0, 0,0, 0,0, 0,0/	CI0ASTG	3
DATA DELZ, HOCURV, #INOUX,#INOUK,IFG,IITH,IIPS	GUL	39
X /0,0, 0,0, 0,1, 0,1, 1, 0, 0/	GUL	40
DATA DUUT,DIR,XPUS,YPOS,YUUT,YIN/ 6*0,0/	SOAPH	16
DATA DIBEAM,OVRLAP, DXXR, UYTH, MAXIT, AVCUSM /4*0,0,1,0,0/	GUL	42
DATA CUSMF/1./	CYCLEY	2
DATA MADPLT/0,0/	GUL	43
DATA ALFA,SCP,T,HMU,ZLEN,NSTEPS,INPT,NPRUP,AXIAL/5*0,0,1,1,0,0,0/	GUL	44
DATA DT /0,0/	GUL	45
DATA IHEAD, IWHITE, IUND, IADD /0,0,1,0/	GUL	46
DATA TRANS, XMAG /1,0,1,0/	GUL	47
DATA NBEAM,AWL /0,0,0/	GUL	48
DATA NGNU /2/	GUL	49
DATA ALPHAM,CUMMIR,ALPHAG,HMUGAS,TAU,TIN,HEFMIR,CUNGAS	GUL	50
X/6*0,0,1,0,0,0/	GUL	51
DATA CAPH,EXPAND,RDC /30...INUE..0,0/ , DISP,TILT/0,0,0/	GUL	52
DATA DELZH, DELZTH, #INOUX, #INOUK	GUL	53
X /0,0, 0,0, 0,1, 0,1/	GUL	54
DATA DSM, REMOVE, PHIAHB	GUL	55
X /0,0, .THUE, , 0,0 /	GUL	56
DATA DIM,XSPC,YSPC,#IDTH,THETA,NSPD/10,14*0,0...423,-120..	CURR2	6
X 0..4*0,0,2/	CURR2	7
NAMLIST/ CUNTL / IFLW,GNOTE,IPLTS	GUL	57
C	GUL	58
C	GUL	59
C	GUL	60
C	GUL	61
C	GUL	62
C	GUL	63
C	GUL	64
C	GUL	65
C	GUL	66
C	GUL	67
C	GUL	68
C	GUL	69
C	GUL	70
C	GUL	71
C	GUL	72
C	GUL	73
C	GUL	74
C	GUL	75
C	GUL	76
C	GUL	77
C	GUL	78
C	GUL	79
C	GUL	80
C	SOU77CY1	165
C	GUL	81
C	GUL	82
C	GUL	83
C	GUL	84
C	GUL	85
C	GUL	86
C	GUL	87

IFLOW CONTROLS THE FLOW OF CALCULATIONS THROUGH GUL

- = 1 CAVITY ELEMENT, HEADS CAVTY1,CAVITY2
- = 2 MIRROR ELEMENT, HEADS MIRM
- = 3 VAMP ELEMENT, HEADS PHUPUT
- = 4 APERTURE ELEMENT, HEADS APTUM
- = 5 THERMAL BLOOMING, HEADS BLOOM
- = 6 INTERPOLATE FIELD OVER SMALLER AREA, HEADS CUTOUT
- = 7 TEST FOR CONVERGENCE OF ITERATION, NO INPUT
- = 8 PLOT FIELD DISTRIBUTION, HEADS TITLE
- = 9 RETURN CONTROL TO CALLING PROGRAM, NO INPUT
- = 10 READ AND/OR WRITE CU ON DISK, HEADS DISKII
- = 11 AERO #INOUX R.M.S. PHASE MODEL, NO INPUT
- = 12 SCALING ROUTINE...MULTIPLIES ENTIRE FIELD, HEADS MULT
- = 13 FLIPS THE FIELD ABOUT THE Y-AXIS, NO INPUT
- = 14 SINUSOIDAL DENSITY VARIATIONS, HEADS SINUEN
- = 15 REGNUS FIELD TO LANGEN SIZE, HEADS REGNID
- = 16 CU PUNCHED ON CANUS, NO INPUT
- = 17 MIRROR THERMAL BL MODEL, HEADS THML
- = 18 SPIDER ROUTINE, HEADS SPIDH
- = 19 AXICN ROUTINE, HEADS AXICUN
- = 20 PROPAGATE IN H-THETA SPACE, HEADS HPNUP
- = 21 REMOVES ON ADUS BACK BEAM CENTER, HEADS CENTER
- = 22 FLIPS THE BEAM ABOUT THE X-AXIS, NO INPUT

IPLTS IS THE PRINTER PLOT SELECTION. IPLTS=ABCDE #WHERE A=1 SELECTS H-THETA PLOTS, B=1 SELECTS ISO INTENSITY PLOT, C=1 SELECTS X AXIS PLOT, D=1 SELECTS DIAGONAL PLOT, AND E=1 SELECTS Y AXIS PLOT., EXAMPLE---IPLTS=1001 SELECTS ISO INTENSITY AND Y AXIS PLOTS IN X-Y COORDINATES.

NAMLIST /CAVTY1/ NCAVNU,ILR,NSTE,NPLT,ZPHUPI,ZPHUPU

C		GOL	86
C	NCAVNU IS THE NUMBER ASSIGNED TO CAVITY FOR IDENTIFICATION	GOL	89
C	ILR INDICATES DIRECTION OF FIELD THROUGH CAVITY	GOL	90
C	= -1 RIGHT TO LEFT	GOL	91
C	= +1 LEFT TO RIGHT	GOL	92
C	NSTE CONTROLS TYPE OF VAMP CODE BETWEEN SEGMENTS	GOL	93
C	= 1 CONSTANT MESH WITH SETUP	GOL	94
C	= 2 VARIABLE MESH WITH SETUP (EXITS VAMP AT END OF ELEMENT)	GOL	95
C	= 3 VARIABLE MESH WITH SETUP (REMAINS IN VAMP)	GOL	96
C	= 4 USE EXISTING PROPAGATING MATRIX (EXITS VAMP)	GOL	97
C	= 5 USE EXISTING PROPAGATING MATRIX (REMAINS IN VAMP)	GOL	98
C	NPLT CONTROLS INTERMEDIATE PRINTOUT FOR CAVITY	GOL	99
C	= 0 NO PRINTOUT	GOL	100
C	= 1 PRINT FIELD BEFORE AND AFTER GAIN, AND GAIN CO-EFF	GOL	101
C	ZPHOPI IS PROPAGATION DISTANCE FROM PREVIOUS OPT. ELEMENT TO CAV.	GOL	102
C	ZPHOPU IS PROPAGATION DISTANCE FROM CAV. TO NEXT OPTICAL ELEMENT	GOL	103
C		GOL	104
C	NAMLIST/MIROR/ANGXX,ANGYY,RAUC,UIAOUT,UIAIN,XMPOS,YMPOS,RMIR,	GOL	105
C	X DELTA,DISTF,DOUTY,DINY,HANULS,PHIAST	CIOASTG	4
C		GOL	107
C	ANGXX IS TILT IN X-DIRECTION - RADIANS (WRT OPT. AXIS)	GOL	108
C	ANGYY IS TILT IN Y-DIRECTION - RADIANS (WRT OPT. AXIS)	GOL	109
C	RAUC IS RADIUS OF CURVATURE OF SPHERICAL MIRROR	GOL	110
C	UIAOUT IS OUTSIDE DIAMETER OF MIRROR	GOL	111
C	UIAIN IS INSIDE DIAMETER OF MIRROR	GOL	112
C	XMPOS IS X-DISPLACEMENT OF MIRROR FROM OPTICAL AXIS	GOL	113
C	YMPOS IS Y-DISPLACEMENT OF MIRROR FROM OPTICAL AXIS	GOL	114
C	RMIR IS REFLECTIVITY OF MIRROR	GOL	115
C	DELTA IS CENTER-TO-EDGE DISTORTION FACTOR (CM)	GOL	116
C	DISTF IS MIRROR DIST. FACTOR (DEFLECTION=DISIF*(1.0-RMIR))	GOL	117
C	HANULS IS OUTSIDE RADIUS OF ANNULAR BEAM (IF APPLICABLE)	GOL	118
C	DOUTY FLAGS THE TYPE OF APERTURE APPLIED -	SQAPH	18
C	.EU. 0 - CIRCULAR APERTURE DEFINED AS ABOVE	SQAPH	19
C	.NE. 0 - RECTANGULAR APERTURE, UIAOUT HIGH (X) BY DOUTY WIDE	SQAPH	20
C	DINY IS SIMILAR TO DOUTY FOR INSIDE DIMENSIONS	SQAPH	21
C	PHIAST IS THE ANGLE OF INCIDENCE OF THE BEAM --- DEGREES	CIOASTG	5
C		GOL	119
C	NAMLIST/ PHOPGT / DELZ, RUCUMV,WINOXX,WINOYY,IIFG,IITR,IIPS	GOL	120
C		GOL	121
C	DELZ IS PROPAGATION DISTANCE	GOL	122
C	RUCUMV IS RADIUS OF CURVATURE OF PHASE FRONT	GOL	123
C	IF ABS(INCUMV) LT 0.5 USE RAUCOH OF PREVIOUS MIRROR	GOL	124
C	WINOXX IS X-SPACE DATA WINDOW FOR FFT	GOL	125
C	WINOYY IS Y-SPACE DATA WINDOW FOR FFT	GOL	126
C	IIFG IS A VAMP CONTROL PARAMETER	GOL	127
C	= 1 FOR CONSTANT MESH	GOL	128
C	= 2 FOR VARIABLE MESH	GOL	129
C	IITR IS ANOTHER VAMP CONTROL PARAMETER	GOL	130
C	= 0 NO INVERSE TRANSFORM	GOL	131
C	= 1 INVERSE TRANSFORM BACK TO REAL SPACE	GOL	132
C		GOL	133
C	IIPS IS FOR CONNECTION OF PLANE AND SPHERICAL PHASE FRONTS	GOL	134
C	= 0 NO CORRECTION	GOL	135
C	= 1 PLANAR CORRECTION ONLY	GOL	136
C	= 2 QUADRATIC CORRECTION ONLY (NOT OPERATIONAL)	GOL	137
C	= 3 BOTH	GOL	138
C		GOL	139
C	NAMLIST /APTUN/ DOUT,DIIN,APUS,YPOS,YOUT,YIN	SQAPH	22
C		GOL	141
C	DOUT IS OUTSIDE DIAMETER OF APERTURE	GOL	142
C	DIIN IS INSIDE DIAMETER OF APERTURE	GOL	143
C	APUS IS X-DISPLACEMENT OF APERTURE FROM OPTICAL AXIS	GOL	144
C	YPOS IS Y-DISPLACEMENT OF APERTURE FROM OPTICAL AXIS	GOL	145
C	YOUT FLAGS THE TYPE OF APERTURE APPLIED -	SQAPH	23
C	.EU. 0 - CIRCULAR APERTURE DEFINED AS ABOVE	SQAPH	24
C	.NE. 0 - RECTANGULAR APERTURE, DOUT HIGH (X) BY YOUT WIDE (Y)	SQAPH	25
C	YIN IS SIMILAR TO YOUT FOR INSIDE DIMENSIONS	SQAPH	26
C		GOL	146
C	NAMLIST /CUTOUT/ DIBEAM,OVNLAP,DIRX,DIRY,MAXIT,AVCUSH,CUSMF	CYCLE9	3
C	CUSMF=1. FOR NORMAL LOADED RESONATOR CUTOUT	CYCLE9	4
C	CUSMF=0. AVOIDS WRITING FIELD ON B AND AVOIDS NORM. FIELD UNLOADED	CYCLE9	5

C			GUL	148
C		DIBEAM IS THE DIAMETER OF BEAM FOR NEXT ITERATION	GUL	149
C		UVNLAP IS: UCALC=UVNLAP*DIBEAM	GUL	150
C		DXR IS POSITION OF ITERATIVE BEAM REL. TO OPTICAL AXIS	GUL	151
C		DYYN IS THE SAME	GUL	152
C		MAXIT IS THE MAX NUMBER OF ITERATIONS	GUL	153
C			GUL	154
C		AVCUM AVERAGES PREVIOUS NEXT ITERATION GUESS IN THE HOPE	GUL	155
C		OF RAPID CONVERGENCE... = 0 NO AV, =.5 ITS HALF AND HALF	GUL	156
C			GUL	157
C		NAMelist/ PLOT / TITLE , HAULPT	GUL	158
C		TITLE IDENTIFIES THE POSITION OF EACH STATION PLOTTED	GUL	159
C		HAULPT CONTROLS THE TYPE OF PLOT	GUL	160
C		= 0.0 FOR X,Y PLOTTING (X-AXIS, Y-AXIS, DIAGONAL)	GUL	161
C		= 1.0 FOR RADIAL PLOTTING AT VARIOUS THETAS	GUL	162
C			GUL	163
C		NAMelist/ BLOOM / ALFA,SCP,T,RMU,ZLEN,NSTEPS,INPT,NPROP,AXIAL,DT	GUL	164
C			GUL	165
C		ALFA = MEDIUM ABSORPTION COEFFICIENT, CM-1	GUL	166
C		SCP = MEDIUM SPECIFIC HEAT, J/GM-DEG K	GUL	167
C		T = MEDIUM TEMPERATURE, DEG K	GUL	168
C		RMU = MEDIUM DENSITY, GM/CM3 (UM TRANSVERSE VEL. IF .GT. 1.)	GUL	169
C		ZLEN = MEDIUM THICKNESS ALONG OPTICAL AXIS	GUL	170
C		NPROP = PROPAGGATION PARAMETER...SAME AS NSTE IN CAVITY	GUL	171
C		NSTEPS = NUMBER OF ELEMENTS IN SUBSYSTEM, .GE. 1	GUL	172
C		INPT = .NE. 0 FOR INTERMEDIATE FIELD PLOTS	GUL	173
C		AXIAL = AXIAL VELOCITY (CM/SEC) IF GT 0, USES AXIAL BLOOMING	GUL	174
C		DT = BEAM ON TIME FOR THERMAL BUY LAYER GROWTH IN TRANSIENT	GUL	175
C		BLOOMING CALCS. IF DT GT 0, USES TRANSIENT BLOOMING	GUL	176
C			GUL	177
C		NAMelist/ DISKIT / IHEAD, IWRITE, IOHD , IADD	GUL	178
C		IHEAD IS THE DISK NUM TO BE READ OFF OF...IF=0...DUNGT HEAD	GUL	179
C		IWRITE IS THE DISK # TO BE WRUTE ON =0...DUNGT WHITE	GUL	180
C		IOHD IS THE ORDER = 1, HEAD FIRST	GUL	181
C		=-1, WHITE FIRST	GUL	182
C		IADD = 1 UPDATES IWRITE BY 1 FOR SUCCESSIVE ITERATIONS	GUL	183
C			GUL	184
C		NAMelist / MULT / TRANS, AMAG	GUL	185
C		TRANS IS TRANSMISSION OF ELEMENT	GUL	186
C			GUL	187
C		NAMelist / SINLEN / NBEAM, AXL	GUL	188
C		NBEAM IS THE NUMBER OF CYCLES PER X-CALCULATED REGION	GUL	189
C		AXL IS THE AMP/WL OF THE SINUSOIDAL VARIATIONS	GUL	190
C			GUL	191
C		NAMelist / HEGRID/ NGRU	GUL	192
C		NGRU IS NO. OF FIELD POINTS ACROSS HEGRIDDED UCAL	GUL	193
C			GUL	194
C			GUL	195
C		NAMelist / THML/ALPHAM,CONMIR,ALPHAG,RMUGAS,TAU,TIN,REFMIR,	GUL	196
C		XCONGAS	GUL	197
C			GUL	198
C		THML IS THE NAMelist FOR BOUNDARY LAYER THERMAL LENS CALCULATIONS	GUL	199
C		ALPHAM= MIRROR DIFFUSIVITY CMSQ/SEC	GUL	200
C		CONMIR= MIRROR THERMAL CONDUCTIVITY WATTS/CM SEC	GUL	201
C		ALPHAG= THERMAL DIFFUSIVITY OF GAS HEATED BY MIRROR CMSQ/SEC	GUL	202
C		CUNGAS= THERMAL CONDUCTIVITY OF GAS HEATED BY MIRROR WATT/CM-SEC	GUL	203
C		RMUGAS= DENSITY OF GAS HEATED BY MIRROR GM/CC	GUL	204
C		TAU = BEAM ON TIME FOR BOUNDARY LAYER GROWTH SEC	GUL	205
C		TIN = INITIAL TEMPERATURE OF GAS & MIRROR DEG K	GUL	206
C		REFMIR= MIRROR REFLECTIVITY (OBTAINED FROM MIRROR INPUT)	GUL	207
C		ETHERMLMAY BE APPLIED AFTER ANY MIRROR TO ALTER THE GAIN - PHASE	GUL	208
C		DUE TO HEATING OF THE QUIESCENT BOUNDARY LAYER ADJACENT TO THE	GUL	209
C		MIRROR SURFACE.	GUL	210
C			GUL	211
C		NAMelist/ SPIDR / NSPU,WIDTH,THETA,ASPC,YSPL,DIM	GUL	212
C		NSPU = NUMBER OF SPOKS IN SPIDR (MAX=6)	GUL	213
C		WIDTH = WIDTH OF SPOKS IN SPIDR	GUL	214
C		THETA = ANGLE OF INDIVIDUAL SPOKS OF SPIDR	GUL	215
C			GUL	
C		ASPC = X-LOCATION OF CENTER OF SPIDR	GUL	216
C		YSPL = Y-LOCATION OF CENTER OF SPIDR	GUL	217
C		DIM = HUB DIAMETER	GUL	218
C			GUL	219

```

C      NAMELIST / AICON / CAPH,EXPAND,MUC,DISP,TILT          GDL      220
C      CAPH IS THE OUTSIDE RADIUS OF THE ANNULAR EXTRACTION BEAM      GDL      221
C      EXPAND EQ. .TRUE. MEANS THE BEAM IS GOING FROM CIRCULAR TO    GDL      222
C      ANNULAR IN CROSS-SECTION                                       GDL      223
C      RUC = RADIUS OF CURVATURE OF THE FIELD IN PHYSICAL SPACE      GDL      224
C      DISP = DISPLACEMENT OF AICON FROM CENTER ALONG X-AXIS       GDL      225
C      TILT = ANGLE(RADIANS) OF AICON TILT FROM DIRECTION OF PROP.   GDL      226
C                                                                    GDL      227
C      NAMELIST/ RPROP / DELZH,DELZTH,WINDUX,WINDUK           GDL      228
C                                                                    GDL      229
C      DELZH IS PROPAGATION DISTANCE FOR THE RADIAL COORDINATE      GDL      230
C      DELZTH IS PROPAGATION DISTANCE FOR THE ANGULAR COORDINATE     GDL      231
C      *** DELZM .NE. DELZTH MEANS YOU ARE MAKING AN EQUIVALENT *** GDL      232
C      *** COLLIMATED BEAM PROPAGATION STEP IN H-THETA COORDINATES ** GDL      233
C      WINDUX IS X-SPACE DATA WINDUX FOR FFT                       GDL      234
C      WINDUK IS K-SPACE DATA WINDUK FOR FFT                      GDL      235
C                                                                    GDL      236
C      NAMELIST/ CENTER / DSM,REMOVE,PHIAMB                   GDL      237
C                                                                    GDL      238
C      DSM IS THE DIAMETER TO BE REMOVED AND LATER ADDED TO THE MAINBEAM GDL      239
C      REMOVE FLAGS THE ACTION -                                     GDL      240
C      .TRUE. IF THE CENTER PORTION OF THE BEAM IS TO BE REMOVED    GDL      241
C      .FALSE. IF THE REMOVED PORTION IS TO BE ADDED BACK TO THE BEAM GDL      242
C      PHIAMB IS AN ARBITRARY PHASE CHANGE ADDED TO THE CENTRAL PORTION GDL      243
C      .....,.....C                                          GDL      244
C                                                                    GDL      245
C      IF (IFLAG.NE.0) GO TO 4/52                                   GDL      246
C      CALL CPUTIM(ISTRT)                                          GDL      247
C      IFLAG=1                                                    GDL      248
C      RAPH=0.0                                                    GDL      249
C      SPPH=1.E70                                                 GDL      250
C      CPCNT=0.0                                                  GDL      251
C      MSTEP=0                                                    GDL      252
C      WHY = .TRUE.                                              GDL      253
C      KAUTU = 0                                                  GDL      254
C      NIT = NITER                                               GDL      255
C      ICNTL=0                                                    GDL      256
C      ANGX=0.                                                    GDL      257
C      ANGY=0.                                                    GDL      258
C      CALL ZERU(ICAV,NCT)                                         GDL      259
C      DO 173 IZERU=1,16                                          GDL      260
C      173 ICAVZ(IZERU)=0                                          GDL      261
C      CALL ZERU(GNOT(1,1),GNOT(50,20))                          GDL      262
C      DO 174 IZERO=1,20                                          GDL      263
C      DO 174 JZERO=1,50                                          GDL      264
C      174 GNOT(JZERO,IZERU) = 0.                                  GDL      265
C      DO 3 I6=1,10                                               GDL      266
C      3 SAVE(I6)=1.                                              GDL      267
C      NUB = NPIS*NPY                                             GDL      268
C      .....,.....C                                          GDL      269
C      BEGIN DIRECTION OF OPTICAL CALCULATIONS                    GDL      270
C      C1000 CALL ZERU(GNOTE(1),GNOTE(20))                        GDL      271
C      1000 DO 176 IZERO=1,20                                     GDL      272
C      176 GNOTE(IZERU)=0.                                        GDL      273
C      HEAD(IN,CUNTHL)                                           GDL      274
C      IGATE = 0                                                  GDL      275
C      HEAD(IN,1243) GNOTE                                       GDL      277
C      1243 FUMMA1(2044)                                          GDL      278
C      ICNTL=ICNTL+1                                             GDL      279
C      IPLTS(ICNTL) = IPLTS                                       GDL      280
C      DO 802 I=1,20                                              GDL      281
C      GNOT(ICNTL,I)=GNOTE(I)                                     GDL      282
C      802 CONTINUE                                              GDL      283
C      WRITE(6,801)(GNOT(ICNTL,I),I=1,20)                        GDL      284
C      801 FUMMA1(1//IX,30(JH**00))/5X,2044//IX,30(JH**00))     GDL      285
C      CALL CPUTIM(INCH)                                          GDL      286
C      TIME=(ISTRT-INCH)/100.                                     GDL      287
C      ISTRT=INCH                                                GDL      288
C      IF(NITER.EQ.0.0)WRITE(6,1002)TIME                         GDL      289
C      C1002 FUMMA1(1//20X,2/HCPU TIME SINCE LAST CUNTHL=,FB,2//) GDL      290
C      ITM = ITM+1                                               GDL      291
C      INIT = .TRUE.                                             GDL      292
C      IGOL(ITM) = IFLW                                          GDL      292

```

C	IFLOW = /1/ 2/ 3/ 4/ 5/ 6/ 7/ 8/ 9/ 10/ 11/ 12/ 13/ 14/ 15/	GUL	293
	GO TO (10,20,30,40,50,60,70,80,900,100,340,350,360,420,150	GUL	294
C	/16 /17 / 18/ 19/ 20/ 21/	GUL	295
	X,160,170,180,190,200,210,365),IFLOW	SUQ77CY1	167
C	ENTRY AUTO(ABC,IB)	GUL	297
4752	HEAD (IB) (CU(IZ),IZ=1,NOB),X,DOWN,WWW	GUL	298
	REWIND IB	GUL	299
	KAUTO = 1	GUL	300
	NIT = 0	GUL	301
	DMX = ABC(1,2,1)	GUL	302
	DMY = ABC(2,2,1)	GUL	303
	NITER = 0	GUL	304
	WMY = .THUE.	GUL	305
C	GUL	306
C	NESTANT POINT FOR SECOND AND SUBSEQUENT ITERATIONS OF A RESONATOR	GUL	307
99	NCT = 0	GUL	308
	ICNTL=0	GUL	309
	INIT=.FALSE.	GUL	310
	ANGX=0.	GUL	311
	ANGY=0.	GUL	312
C	CALL ZERO(ICAV,IOK)	GUL	313
	DO 177 IZERO=1,13	GUL	314
177	ICAVZ(IZERO) = 0	GUL	315
	IWMA = 0	GUL	316
98	IWMA = IWMA + 1	GUL	317
	ICNTL=ICNTL+1	GUL	318
	IPLTS = IPLTS(ICNTL)	GUL	319
	IGATE = 0	GUL	320
	IFLOW=IGUL(IWMA)	GUL	321
	WRITE(6,801)(GNOT(ICNTL,1),I=1,20)	GUL	322
C	IFLOW = /1/ 2/ 3/ 4/ 5/ 6/ 7/ 8/ 9/ 10/ 11/ 12/ 13/ 14/ 15/	GUL	323
	GO TO (10,20,30,40,50,60,70,80,900,100,340,350,360,420,150	GUL	324
C	/16/17/ 18/ 19/ 20/ 21/	GUL	325
	X,160,160,180,190,200,210,365),IFLOW	SUQ77CY1	168
	STOP	GUL	327
C	GUL	328
C	CUT OUT FIELD CENTER AND SAVE OR ADD TO CURRENT FIELD	GUL	329
210	ICUT = ICUT+1	GUL	330
	IF(.NOT. INIT) GO TO 212	GUL	331
	HEAD(IN,CENTER)	GUL	332
	USMM(ICUT) = USM/2.	GUL	333
	RMV(ICUT) = REMOVE	GUL	334
	PHIA(ICUT) = PHIAHB	GUL	335
212	IF(.NOT.RMV(ICUT)) GO TO 216	GUL	336
	WRITE (6,214) USMM(ICUT)	GUL	337
214	FORMAT(/29H THE BEAM CENTER (RADIUS = .F6.3,20H) HAS BEEN REMOV	GUL	338
	XED //)	GUL	339
	GO TO 219	GUL	340
216	WRITE (6,217) USMM(ICUT),PHIA(ICUT)	GUL	341
217	FORMAT(/29H THE BEAM CENTER (RADIUS = .F6.3,58H) HAS BEEN ADDED	GUL	342
	X BACK TO THE BEAM WITH A PHASE CHANGE OF .F7.4//)	GUL	343
219	CALL FIELDS(USMM(ICUT),RMV(ICUT),PHIA(ICUT))	GUL	344
	IGATE = 1	GUL	345
	GO TO 3623	GUL	346
C	GUL	347
C	PROPAGATE UNWOLLED ANNULUS	GUL	348
200	INSTP = IRSTP+1	GUL	349
	NPYP1=NPY+1	GUL	350
	IF(.NOT. INIT) GO TO 232	GUL	351
	HEAD(IN,NPROP)	GUL	352
	ABC(1,INSTP,8) = DELZH	GUL	353
	ABC(2,INSTP,8) = DELZTH	GUL	354
	IF(ABC(2,INSTP,8).EQ.0.0) ABC(2,INSTP,8) = DELZH	GUL	355
	ABC(3,INSTP,8) = WINDOX	GUL	356
	ABC(4,INSTP,8) = WINDOK	GUL	357
232	WRITE (6,234) (ABC(1,INSTP,8),1,1),ANGX,ANGY	GUL	358
234	FORMAT(//59H DELZH DELZTH WINDOX WINDOK ANGA	GUL	359
	X ANGY. / 6F10.4//)	GUL	360
	CALL NSTEP(ABC(1,INSTP,8),ABC(2,INSTP,8),ABC(3,INSTP,8),ABC(4,	GUL	361
	INSTP,8),ANGX,ANGY,Y)	GUL	362
	CALL POWH(CU,X,NPTS,NPYP1)	GUL	363
	IGATE = 1	GUL	364
	GO TO 3623	GUL	365

C	GUL	366
C	APPLY AXICUN	GUL	367
190	IAX=IAX+1	GUL	368
	IF(.NOT.INIT) GO TO 191	GUL	369
	HEAD(IN,AXICUN)	GUL	370
	CPH(IAX)=CAPH	GUL	371
	XPNO(IAX)=EXPHNU	GUL	372
	RCURVE(IAX)=ROC	GUL	373
	TLT(IAX) = TILT	GUL	374
	USP(IAX) = USP	GUL	375
191	CALL AXICN(CPH(IAX),XPNO(IAX),RCURVE(IAX),USP(IAX),TLT(IAX),Y)	GUL	376
	GO TO 999	GUL	377
C	GUL	378
C	APPLY SPIDER OSCURATION TRANSMISSION FUNCTION TO THE COMPLEX	GUL	379
C	FIELD	GUL	380
180	HEAD(IN,SPIDN)	GUL	381
	WRITE(6,181)WIDTH,NSPD,ASPC,YSPC,DIM,(THETA(ISPD),ISPD=1,NSPD)	GUL	382
181	FORMAT(20H0 SPIDEN MODEL APPLIED: /,15H STRUT WIDTH =,G12.3,15H	GUL	383
	INU. OF STRUTS=,I3.12H X-Y LENGTH=,G12.4,1H,,G12.4,	GUL	384
	215H HUB DIAMETER =,G12.4/4H THETAS =,G12.4)	GUL	385
	NSPD = MIN0(NSPD,0)	GUL	386
182	CALL SPIDEN(WIDTH,THETA,NSPD,ASPC,YSPC,DIM)	GUL	387
	IGNAL=0	GUL	388
	GO TO 999	GUL	389
C	GUL	390
C	WRITE COMPLEX FIELD ON PUNCH CARDS	GUL	391
160	WRITE(6,163)	GUL	392
163	FORMAT(36H0 CU HAS BEEN WRITTEN ON PUNCH CARDS)	GUL	393
	WRITE(4,164) (GNUT(ICNTL,I),I=1,20)	GUL	394
164	FORMAT(20A4)	GUL	395
	DO 161 J=1,NPY	GUL	396
	DO 161 I=1,NPTS,2	GUL	397
	IREF = (J-1)*NPTS	GUL	398
	DUM1=REAL(CU(IREF+I))	GUL	399
	DUME1=AIMAG(CU(IREF+I))	GUL	400
	DUM2=REAL(CU(IREF+1+I))	GUL	401
	DUME2=AIMAG(CU(IREF+1+I))	GUL	402
161	WRITE(4,162)X(I),X(J),DUM1,DUME1,X(I+1),X(J),DUM2,DUME2	GUL	403
162	FORMAT(2F8.2,2E12.4,2F8.2,2E12.4)	GUL	404
	IGATE = 1	GUL	405
	GO TO 3623	GUL	406
C	GUL	407
C	APPLY SINUSOIDAL PHASE VARIATION TO COMPLEX FIELD	GUL	408
420	IF (.NOT.INIT) GO TO 421	GUL	409
	READ (IN,SINUEN)	GUL	410
421	WRITE (6,422) NBEAM,AWL	GUL	411
422	FORMAT (/48H SINUSOIDAL DENSITY FIELD APPLIED TO THE BEAM /20H	GUL	412
	X = OF CYCLES PER XCALC ,15,26H AMP/WL OF VARIATIONS =,F7.3)	GUL	413
	AS = 2.*J.141592 * AWL	GUL	414
	AB = 2. * 3.141592 * NBEAM / (NPTS*(X(2)-X(1)))	GUL	415
	DO 423 I=1,NPTS	GUL	416
	CFACTT= CEXP(CMPLX(0., AS * SIN (AB*X(1))))	GUL	417
	DO 423 J=1,NPY	GUL	418
	IJ = I + (J-1)*NPTS	GUL	419
423	CU(IJ) = CU(IJ)*CFACTT	GUL	420
	GO TO 999	GUL	421
C	GUL	422
C	APPLY GOL CAVITY TO COMPLEX FIELD	GUL	423
10	ICAV=ICAV+1	GUL	424
	IF(.NOT. INIT) GO TO 11	GUL	425
	HEAD(IN,CAVTY1)	GUL	426
	IDIH(1,ICAV) = NCAVNO	GUL	427
	IUIH(2,ICAV) = ILH	GUL	428
	IDIR(3,ICAV) = NSTE	GUL	429
	IDIR(4,ICAV) = NPLT	GUL	430
	ZLI(ICAV)=ZPHUPI	GUL	431
	ZLU(ICAV)=ZPHOPO	GUL	432
11	NE=CAV = 0	GUL	433
	NCS = MAX0(IDIR(1,ICAV),NCT)	GUL	434
	IF(NCS.GT.NCT) NE=CAV=1	GUL	435
	NCT = NCS	GUL	436
	WRITE (6,12) IDIH(1,ICAV),IUIH(2,ICAV),IDIR(3,ICAV)	GUL	437

12	FORMAT (///16M CAVITY NUMBER,13,1/M	DIRECTION ,12 ,29H	GUL	438
	X PROPAGATING PARAMETER ,12 /)		GUL	439
	WRITE(6,15) ZLI(1,ICAV), ZLU(1,ICAV)		GUL	440
15	FORMAT(4#H0ADDITIONAL PROPAGATION DISTANCES AT CAVITY ENUS/		GUL	441
	X 1X,4MZLI=,G12.5,0X,4MZLU=,G12.5)		GUL	442
	CALL CAVITY(IDIR(1,ICAV),IDIR(2,ICAV),NEWCAV,INIT,DIR(J,ICAV),IN		GUL	443
	X HESINT, IDIR(4,ICAV),ZLI(1,ICAV),ZLU(1,ICAV))		GUL	444
	IF (IDIR(J,ICAV).LE.3) INT=1		GUL	445
	GO TO 999		GUL	446
C		GUL	447
C	READ ANU/UM WRITE COMPLEX FIELD ON DIRECT ACCESS FILE		GUL	448
100	NUS = NUS + 1		GUL	449
	IF (.NOT. INIT) GO TO 101		GUL	450
	HEAD(IN,DISKIT)		GUL	451
	IUSK(1,NUS) = IHEAD		GUL	452
	IUSK(2,NUS) = IWRITE		GUL	453
	IUSK(3,NUS) = IORD		GUL	454
	IUSK(4,NUS) = IADD		GUL	455
	GO TO 107		GUL	456
101	IHEAD = IUSK(1,NUS)		GUL	457
	IUSK(2,NUS) = IUSK(2,NUS) + IUSK(4,NUS)		GUL	458
	IWRITE = IUSK(2,NUS)		GUL	459
	IORD = IUSK(3,NUS)		GUL	460
107	IF (IHEAD.EQ.0.OR.IORD.EQ.-1) GO TO 102		GUL	461
	HEAD (IHEAD) (CU(12),I2=1,NUB),X,DNA,DNY,NITER		GUL	462
	WRITE(6,105)IHEAD		GUL	463
105	FORMAT(//10X,26MCU HAS BEEN READ FROM UNIT,13//)		GUL	464
	RE=IND IHEAD		GUL	465
102	IF (IWRITE.EQ.0) GO TO 103		GUL	466
	WRITE (IWRITE) (CU(12),I2=1,NUB),X,DNA,DNY,NITER,SAVE		GUL	467
	WRITE(6,106)IWRITE		GUL	468
106	FORMAT(//10X,27MCU HAS BEEN WRITTEN ON UNIT,13//)		GUL	469
	RE=IND IWRITE		GUL	470
103	IF (IHEAD.EQ.0.OR.IORD.EQ.1) GO TO 999		GUL	471
	HEAD (IHEAD) (CU(12),I2=1,NUB),X,DNA,DNY,NITER		GUL	472
	WRITE(6,105)IHEAD		GUL	473
	RE=IND IHEAD		GUL	474
	GO TO 999		GUL	475
C		GUL	476
C	APPLY AERODYNAMIC WINDOW TO COMPLEX FIELD		GUL	477
340	WRITE (6,341)		GUL	478
341	FORMAT (//74M AERU WINDOW MODEL HAS BEEN APPLIED...NMS PHASE DIST		GUL	479
	XOPTION IS THE MODEL /)		GUL	480
	CALL AEROW(CU,NPTS,NPY)		GUL	481
	GO TO 999		GUL	482
C		GUL	483
C	APPLY FIELD SCALING FACTOR		GUL	484
350	MLT=MLT+1		GUL	485
	IF (.NOT. INIT) GO TO 351		GUL	486
	HEAD (IN,MULT)		GUL	487
	ABC(1,MLT,9)=TRANS		GUL	488
	ABC(2,MLT,9)=XMAX		GUL	489
351	WRITE(6,352) ABC(1,MLT,9),ABC(2,MLT,9)		GUL	490
	STRANS = SQRT(ABC(1,MLT,9))/ABC(2,MLT,9)		GUL	491
352	FORMAT (/43M THE FIELD HAS BEEN SCALED BY THE FACTORS ,2F8.3/)		GUL	492
	DO 353 I=1,NUB		GUL	493
353	CU(I) = CU(I)*STRANS		GUL	494
	DO 357 I = 1,NPTS		GUL	495
357	X(I) = X(I) * ABC(2,MLT,9)		GUL	496
	RMIRN = ABC(1,MLT,9)		GUL	497
	IGNAL = 5		GUL	498
	GO TO 999		GUL	499
C		GUL	500
C	MAKE PRINTER PLOTS OF COMPLEX FIELD		GUL	501
80	IPIT=IPIT+1		GUL	502
	IF (.NOT. INIT) GO TO 82		GUL	503
	HEAD (IN,PLOT)		GUL	504
	HEAD (5,1243) TITLE		GUL	505
	DU 83 NU=1,20		GUL	506
83	AMPL(IPIT,NU)=TITLE(NU)		GUL	507
82	WRITE (6,84) (AMPL(IPIT,NU),NU=1,20)		GUL	508

84	FORMAT (I11,J0X,20A4 //)	GUL	509
	IF (RAUPLI.EQ.0.0) CALL IPLUT (11111)	GUL	510
	IF (RAUPLI.NE.0.0) CALL IPLUT (11111)	GUL	511
	IF (.NOT.INIT) GO TO 98	GUL	512
	GO TO 1000	GUL	513
C	GUL	514
C	FLIP THE COMPLEX FIELD ABOUT THE Y-AXIS	GUL	515
360	NP = NPTS / 2	GUL	516
	WRITE (6,361)	GUL	517
361	FORMAT (/52M THE FIELD HAS JUST BEEN FLIPPED ABOUT THE Y-AXIS /)	GUL	518
	DO 362 J=1,NPY	GUL	519
	DO 362 I=1,NP	GUL	520
	I2 = I + (J-1) * NPTS	GUL	521
	I3 = I - I * NPTS * J	GUL	522
	CUD = CU (I2)	GUL	523
	CU(I2) = CU(I3)	GUL	524
362	CU(I3) = CUD	GUL	525
	GO TO 999	GUL	526
C	SQU77CY1	169
365	IF (NPY.NE.NPTS) GO TO 999	SQU77CY1	170
	NP=NPTS/2	SQU77CY1	171
	WRITE(6,366)	SQU77CY1	172
366	FORMAT(/46M THE FIELD HAS BEEN FLIPPED ABOUT THE X-AXIS/)	SQU77CY1	173
	DO 367 I=1,NPTS	SQU77CY1	174
	DO 367 J=1,NP	SQU77CY1	175
	I2=I+(J-1)*NPTS	SQU77CY1	176
	I3=I-NPB-J*NPTS	SQU77CY1	177
	CUD=CU(I2)	SQU77CY1	178
	CU(I2)=CU(I3)	SQU77CY1	179
367	CU(I3)=CUD	SQU77CY1	180
	GO TO 999	SQU77CY1	181
C	GUL	527
C	APPLY MIRROR TRANSMISSION FUNCTION TO THE COMPLEX FIELD	GUL	528
20	MIR = IMIR+1	GUL	529
	IF (.NOT. INIT) GO TO 21	GUL	530
	HEAD(IN,MIR,0)	GUL	531
	ABC(1,IMIR,2) = ANGXA	GUL	532
	ABC(2,IMIR,2) = ANGY	GUL	533
	ABC(3,IMIR,2) = NAUC	GUL	534
	ABC(4,IMIR,2) = DIAOUT/2.	GUL	535
	ABC(5,IMIR,2) = DIAIN/2.	GUL	536
	ABC(6,IMIR,2) = XMPUS	GUL	537
	ABC(7,IMIR,2) = YMPUS	GUL	538
	ABC(8,IMIR,2) = XMIH	GUL	539
	ABC(9,IMIR,2) = DELTA	GUL	540
	ABC(10,IMIR,2) = DISIF	GUL	541
	ABC(11,IMIR,2) = MANULS	GUL	542
	ABC(10,IMIR,4) = UOUTY/2.	SWAPH	27
	ABC(11,IMIR,4) = UINY/2.	SWAPH	28
	ABC(12,IMIR,2) = PHIAT	CIUASTG	6
21	CALL MIRROR(ABC(1,IMIR,2),ABC(2,IMIR,2),ABC(3,IMIR,2),ABC(4,IMIR,2)	GUL	543
	1),ABC(5,IMIR,2),ABC(6,IMIR,2),ABC(7,IMIR,2),ABC(8,IMIR,2),	GUL	544
	2 ABC(9,IMIR,2),ABC(10,IMIR,2),ABC(11,IMIR,2),ABC(10,IMIR,4),	SWAPH	29
	3 ABC(11,IMIR,4),ABC(12,IMIR,2))	CIUASTG	7
	NAPTM=ABC(4,IMIR,2)	GUL	546
	WRITE(6,23) (ABC(IMN,IMIR,2),IMN=1,3),(ABC(IMN,IMIR,2),IMN=6,11)	SWAPH	31
23	FORMAT(///8M ANGXA =,G12.4,8M ANGY =,G12.4,17M RADIUS OF CURV =,G	GUL	548
	X12.4/ *3M SWAPH	SWAPH	32
	X POSITION OF MIRROR A.M.T. OPTICAL AXIS = (,F6.3,IM,*,F6.3,IM) /	GUL	550
	X22M MIRROR REFLECTIVITY =,F6.3,5X/	GUL	551
	X37M MIRROR SPHERICAL DISTORTION FACTOR =,E12.4/	GUL	552
	X37M MIRROR FLUX DEP. DISTORTION FACTOR =,E12.4/	GUL	553
	X37M OUTSIDE RADIUS OF ANNULAR BEAM =,E12.4/)	GUL	554
	IF (ABC(10,IMIR,2).GT.-10.) GO TO 8316	EUIPWH	1
	WRITE(6,3913)	EUIPWH	2
3913	FORMAT(58M EDI LOSS ACCOUNTED FOR IN ASSOCIATED MIRROR CALCULATION	EUIPWH	3
	XS)	EUIPWH	4
	GO TO 3623	EUIPWH	5
8316	CONTINUE	EUIPWH	6
	IGNAL=2	GUL	555
	IF (ABC(4,IMIR,2).LE.0.0.AND.ABC(5,IMIR,2).EQ.0.0)IGNAL=5	GUL	556
	MIRH=ABC(8,IMIR,2)	GUL	557
	GO TO 999	GUL	558

C	*****	GUL	559
C	APPLY TRANSMISSION FUNCTION OF A QUIESCENT THERMAL GRADIENTS	GUL	560
C	NEAR A MIRROR SURFACE	GUL	561
	170 ITHRML = ITHMML * 1	GUL	562
	IF (.NOT. INIT) GO TO 171	GUL	563
	READ (IN, ITHRML)	GUL	564
	ABC(1, ITHMML, 7) = ALPHAM	GUL	565
	ABC(2, ITHMML, 7) = CUNMIH	GUL	566
	ABC(3, ITHMML, 7) = ALPHAG	GUL	567
	ABC(4, ITHMML, 7) = HMUGAS	GUL	568
	ABC(5, ITHMML, 7) = TAU	GUL	569
	ABC(6, ITHMML, 7) = TIN	GUL	570
	ABC(7, ITHMML, 7) = HEFMIH	GUL	571
	ABC(8, ITHMML, 7) = CUNGAS	GUL	572
	171 CALL THEMML (ABC(1, ITHMML, 7), ABC(2, ITHMML, 7), ABC(3, ITHMML, 7), ABC(4,	GUL	573
	1, ITHMML, 7), ABC(5, ITHMML, 7), ABC(6, ITHMML, 7), ABC(7, ITHMML, 7),	GUL	574
	2, ABC(8, ITHMML, 7))	GUL	575
	IGNAL=1	GUL	576
	GO TO 999	GUL	577
C	*****	GUL	578
C	APPLY PROPAGATION ALGORITHM TO COMPLEX FIELD	GUL	579
	30 ISTEP = ISTEP+1	GUL	580
	IF (.NOT. IN1) GO TO 32	GUL	581
	READ (IN, PRUPGT)	GUL	582
	IF (IIPS.GT.1) IIFG=2	GUL	583
	ABC(1, ISTEP, 3) = UELZ	GUL	584
	ABC(2, ISTEP, 3) = HUCURV	GUL	585
	ABC(3, ISTEP, 3) = WINOUX	GUL	586
	ABC(4, ISTEP, 3) = WINOUK	GUL	587
	ABC(5, ISTEP, 3) = IIFG	GUL	588
	ABC(6, ISTEP, 3) = IITH	GUL	589
	ABC(7, ISTEP, 3) = IIPS	GUL	590
	32 IFG = ABC(5, ISTEP, 3) * .001	GUL	591
	IIM = ABC(6, ISTEP, 3) * .001	GUL	592
	IIS = ABC(7, ISTEP, 3) * .001	GUL	593
	WRITE (6, 34) (ABC(1, ISTEP, 3), ISTEP, IIFG, IITH, IIPS, ANGA, ANGY	GUL	594
	34 FORMAT (/// 91M UELZ HAU CURV WINOUX WINOUK IFG	GUL	595
	X ITH IIS ANGA ANGY / 4F10.4, 16.5X, 16.5X, 16.	GUL	596
	X5A, 2F10.5//)	GUL	597
	ICORE = 0	GUL	598
	IF (IFG.LT.-5) GO TO 31	GUL	599
	IF (ABS(ABC(2, ISTEP, 3)).LT.-5) ABC(2, ISTEP, 3) = HAU CURV	GUL	600
	402 CALL STEP (ABC(1, ISTEP, 3), ABC(2, ISTEP, 3), ABC(3, ISTEP	GUL	601
	1, 3), ABC(4, ISTEP, 3), IFG, IITH, IIPS, ANGA, ANGY, 0, ICORE)	GUL	602
	IF (ICORE.EQ.0) INT = 1	GUL	603
	MSTEP=1	GUL	604
	GO TO 999	GUL	605
	31 IF (INT.EQ.0) WRITE (6, 319)	GUL	606
	319 FORMAT (50M ENTERING CORE BEFORE STEP CALLED; CALCULATIONS STOPPED	GUL	607
	X)	GUL	608
	IF (INT.EQ.0) STOP	GUL	609
C	CALL CORE (ABC(1, ISTEP, 3), IIM, 0)	GUL	610
	ICORE=1	GUL	611
	GO TO 402	GUL	612
C	*****	GUL	613
C	APPLY APERTURE TRANSMISSION FUNCTION TO COMPLEX FIELD	GUL	614
	40 IAP = IAP+1	GUL	615
	IF (.NOT. INIT) GO TO 41	GUL	616
	HEAD (IN, APTUM)	GUL	617
	ABC(1, IAP, 4) = UOUT/2.	GUL	618
	ABC(2, IAP, 4) = DIN/2.	GUL	619
	ABC(3, IAP, 4) = XPOS	GUL	620
	ABC(4, IAP, 4) = YPOS	GUL	621
	ABC(5, IAP, 4) = YOUT/2.	SWAPH	33
	ABC(6, IAP, 4) = YIN/2.	SWAPH	34
	41 IF (DOUT.LT.0.0.AND.DIN.LT.0.0)	GUL	622
	ICALL SLIVER (ABC(1, IAP, 4), ABC(2, IAP, 4), ABC(3, IAP, 4), ABC(4, IAP, 4))	GUL	623
	IF (DOUT.GE.0.0.AND.DIN.GE.0.0) CALL APPTH (ABC(1, IAP, 4), ABC(2, IAP, 4	SWAPH	35
	X), ABC(3, IAP, 4), ABC(4, IAP, 4), ABC(5, IAP, 4), ABC(6, IAP, 4))	SWAPH	36
	IF (DOUT.GT.0.0.AND.DIN.GE.0.0) HAPTH=ABC(1, IAP, 4)	GUL	626
	IGNAL=4	GUL	630
	GO TO 999	GUL	631

C	*****	GUL	632
C	APPLY THERMAL BLOOMING TRANSMISSION FUNCTION TO COMPLEX FIELD	GUL	633
50	IDK = IDK+1	GUL	634
	IF (.NOT. INIT) GO TO 51	GUL	635
	READ(IN,BLOOM)	GUL	636
	ABC(1, IDK, 5) = ALFA	GUL	637
	ABC(2, IDK, 5) = SCP	GUL	638
	ABC(3, IDK, 5) = T	GUL	639
	ABC(4, IDK, 5) = HMO	GUL	640
	ABC(5, IDK, 5) = ZLEN	GUL	641
	ABC(6, IDK, 5) = NSTEPS	GUL	642
	ABC(7, IDK, 5) = INPT	GUL	643
	ABC(8, IDK, 5) = NPHOM	GUL	644
	ABC(9, IDK, 5) = AXIAL	GUL	645
	ABC(10, IDK, 5) = DT	GUL	646
51	NSTEPS = ABC(6, IDK, 5) * .0001	GUL	647
	INPT = ABC(7, IDK, 5) * .0001	GUL	648
	I2ZT = ABC(8, IDK, 5) * .0001	GUL	649
	CALL TBLUOM(ABC(1, IDK, 5), ABC(2, IDK, 5), ABC(3, IDK, 5), ABC(4, IDK, 5),	GUL	650
	ABC(5, IDK, 5), NSTEPS, INPT, I2ZT, ABC(9, IDK, 5), ABC(10, IDK, 5))	GUL	651
	GO TO 999	GUL	652
C	*****	GUL	653
C	INTERPOLATE FEEDBACK FIELD FROM RESONATOR MODE FOR USE IN NEXT	GUL	654
C	ITERATION	GUL	655
60	IF (.NOT. INIT.AND..NOT.DMY) GO TO 61	GUL	656
	IF (.NOT. INIT) GO TO 67	GUL	657
	HEAD(IN,CUTOOUT)	GUL	658
	ABC(1, 1, 1) = DIBEAM	GUL	659
	ABC(2, 1, 1) = UVHLAP	GUL	660
	ABC(3, 1, 1) = DAXM	GUL	661
	ABC(4, 1, 1) = DYYM	GUL	662
	ABC(5, 1, 1) = AVCUSH	GUL	663
	IGUL(99) = IABS(MAX11)	GUL	664
67	DCIBM = ABC(2, 1, 1) * ABC(1, 1, 1) / 2.	GUL	665
	DIBEAM = ABC(1, 1, 1)	GUL	666
	XDEL = DCIBM / NPTS * 2.	GUL	667
	XK(1) = -DCIBM * XDEL / 2.	GUL	668
	DO 62 IGM=2, NPTS	GUL	669
62	XK(IGM) = XK(IGM-1) * XDEL	GUL	670
	TXY(1) = X(2) - X(1)	GUL	671
	TXY(2) = X(2) - X(1)	GUL	672
	TXY(3) = NPY	GUL	673
	TXY(4) = NPTS	GUL	674
	DO 64 MSP=1, NPY	GUL	675
64	TXY(4+MSP) = X(MSP) * DRY	GUL	676
	NPY4 = NPY * 4	GUL	677
	DO 640 MST=1, NPTS	GUL	678
640	TXY(NPY4 + MST) = X(MST) * OMX	GUL	679
61	AVC = ABC(5, 1, 1)	GUL	680
	POWA = 0.	APH27	1
	DX2 = (X(2) - X(1)) / 2.	APH27	2
	UB2 = DCIBM	APH27	3
	DO 621 J=1, NPY	APH27	4
	IF (ABS(X(J)) - DX2 .GT. UB2) GO TO 621	APH27	5
	FCY = 1.0	APH27	6
	IF (ABS(X(J)) * DX2 .LT. UB2) GO TO 627	APH27	7
	FCY = (UB2 - (ABS(X(J)) - DX2)) / DX2 / 2.	APH27	8
627	J1 = (J-1) * NPTS	APH27	9
	DO 620 I=1, NPTS	APH27	10
	IF (ABS(X(I)) - DX2 .GT. UB2) GO TO 620	APH27	11
	FCX = 1.0	APH27	12
	IF (ABS(X(I)) * DX2 .LT. UB2) GO TO 628	APH27	13
	FCX = (UB2 - (ABS(X(I)) - DX2)) / DX2 / 2.	APH27	14
628	IX = J1 * I	APH27	15
	POWA = POWA + CU(IX) * CUNJG(CU(IX)) * FCX * FCY	APH27	16
620	CONTINUE	APH27	17
621	CONTINUE	APH27	18
	POWA = POWA * (X(2) - X(1)) ** 2 / 1000.	APH27	19
	MAAA = 0	GUL	681
	I2=0	GUL	682
	IF (NPTS.NE.NPY) I2=1	GUL	683

```

PWB = 0.
DO 63 MY=1,NPY
YINP = XK(MY) * ABC(4,1,1)
DO 63 MX=1,NPTS
MAAA=MAAA*1
XINP = XK(MX) * ABC(3,1,1)
CALL INTERP(TXY,XINP,YINP,CU,2,CFPL(MAAA),IZ)
63 PWB = PWB + CFPL(MAAA)*CONJG(CFPL(MAAA))
PWB = PWB * (XK(2)-XK(1))**2 / 1000.
FXFLT = SQRT(PWA/PWB)
DO 623 I1 = 1,NOB
623 CFPL(I1) = CFPL(I1)*FXFLT
WRITE (6,624) PWB,PWA
624 FORMAT (/1X,2BFIELD ADJUSTED FROM POWER OF .F8.2,4M TO .F8.2/)
IF (CUSMF.NE.0.) GO TO 5926
IZ=0
DO 5843 I1=1,NPTS
X(I1) = XK(I1)
DO 5843 I2=1,NPY
IZ = IZ + 1
5843 CU(I2) = CFPL(I2)
GO TO 999
5924 CONTINUE
IF (ICAV.GT.0) GO TO 691
FMAX=0.
DO 692 I1=1,NOB
FMAG=CABS(CFPL(I1))
IF (FMAG.LT.FMAX) GO TO 692
FMAX=FMAG
INUM=I1
692 CONTINUE
DO 693 I1=1,NOB
693 CFPL(I1)=CFPL(I1)/FMAX
WRITE(6,694) FMAX
694 FORMAT(/7M CUTOUT FIELD AMPLITUDES HAVE BEEN DIVIDED BY ,
X F8.4//)
691 CONTINUE
WRITE (7) (CU(I2),I2=1,NOB)
HEWIND 7
IF (.NOT.HESTHT.AND.INIT) GO TO 630
HEAD (8) (CFIL2(I2),I2=1,NOB),XDUM,UUM2,UUM3,NDUUM,SAVE
HEWIND 8
630 SUMENH=0.0
ICNT=0
NWTB=NPTS/16
NWTB=NPTS/4
NWTB=NPTS-NWTB
NWTB=NPTS/2
WRITE(6,653)
653 FORMAT(44MOCUTOUT FIELD COMPARISON TO DETERMINE AVGAIN/)
WRITE(6,71)
71 FORMAT(10M POINT .6X.12M CURRENT .4X.12M PREVIOUS .4X.12M
X PERCENT /10M TESTED .6X.12M VALUE .4X.12M VALUE ,
X4X.9M CHANGE//)
ICERS=0
DO 65 IABC=NWTB,NWTC,NWTA
ICNT=ICNT+1
ENHSM=0.
DUM=CABS(CFPL(IABC*(NWTB-1)*NPTS))
DUME=CABS(CFPL2(IABC*(NWTB-1)*NPTS))
IF (.NOT.HESTHT.AND.INIT) DUME=1.0
IF (DUME.NE.0.) ENHSM=1000*DUME/DUME
IF (ABS(ENHSM).GT.0.10) ICERS=1
SUMENH=ENHSM**2*SUMENH
WRITE(6,650) IABC,NWTB,DUM,DUME,ENHSM
650 FORMAT(6M CUSM(.13,1M.12.1M),4X.612.5,4X.612.5,7X.2MF6.2)
65 CONTINUE
IF (ABC(5,1,1).EQ.0. .OR.(NIEN.EQ.0.AND.RAUT0.EQ.0)) GO TO 69
IF (ABC(5,1,1).GE.0.) GO TO 68
ENHSS=SUM(SUMENH/ICNT)
AVC = .8 - ENHSS
IF (ENHSS.GT.0.6) AVC=0.2
IF (ENHSS.LT.0.1) AVC = .7

```

```

APM26 12
GUL 684
GUL 685
GUL 686
GUL 687
GUL 688
APM26 13
APM26 14
APM26 15
APM26 16
APM26 17
APM26 18
APM26 19
APM26 20
CYCLE9 6
CYCLE9 7
CYCLE9 8
CYCLE9 9
CYCLE9 10
CYCLE9 11
CYCLE9 12
CYCLE9 13
CYCLE9 14
GUL 690
GUL 691
GUL 692
GUL 693
GUL 694
GUL 695
GUL 696
GUL 697
GUL 698
GUL 699
GUL 700
GUL 701
GUL 702
GUL 703
GUL 704
GUL 705
GUL 706
GUL 707
GUL 708
GUL 709
GUL 710
GUL 711
GUL 712
GUL 713
GUL 714
GUL 715
GUL 716
GUL 717
GUL 718
GUL 719
GUL 720
GUL 721
GUL 722
GUL 723
GUL 724
GUL 725
GUL 726
GUL 727
GUL 728
GUL 729
GUL 730
GUL 731
GUL 732
GUL 733
GUL 734
GUL 735
GUL 736
SUG77CY1 182
GUL 738
SUG77CY1 183

```

WRITE(6,610)ERRSS,AVC	GUL	740
610 FORMAT(//17X,29HFIELD AVERAGING HAS BEEN USED/10X,10MHMS ERROR=	GUL	741
1 F8.4,5X,12HVCUSM USED=F8.4//)	GUL	742
68 CONTINUE	GUL	743
DO 75 MX=1,NOB	GUL	744
XAN = CABS(CFPL(MX))	SOU77CY1	184
XAOLD = CABS(CFIL2(MX))	SOU77CY1	185
75 CFPL(MX) = CFPL(MX) * (AVC*XAOLD*(1.-AVC)*XAN) / XAN	SOU77CY1	186
69 MY = NITEN+1	GUL	746
HEAD(7) (CU(12),12=1,NOB)	GUL	747
REWIND 7	GUL	748
WRITE(6,663)	GUL	749
663 FORMAT(12X,33HCUNVERGENCE TEST FIELD COMPARISON/)	GUL	750
ICEK=0	GUL	751
SMERH=0.0	GUL	752
ICNT=0	GUL	753
WRITE(6,71)	GUL	754
DO 660 IABC=NWTB,NWTC,NWIA	GUL	755
ICNT=ICNT+1	GUL	756
ENH=0.	GUL	757
DUM=CABS(CU(IABC*(NWTU-1)*NPTS))	GUL	758
DUME=SAVE(ICNT)	GUL	759
SAVE(ICNT)=DUM	GUL	760
IF(.NOT.HESTHT.AND.INIT)DUME=1.0	GUL	761
IF(DUME.NE.0.) ENH=(DUM-DUME)/DUME	GUL	762
IF(ABS(ENH).GT.0.02)ICEK=1	GUL	763
SMERH=ENH*2+SMERH	GUL	764
WRITE(6,661)IABC,NWTU,DUM,DUME,ENH	GUL	765
661 FORMAT(4M CU(,13,1M,12,1M),4X,6I2.5,4X,6I2.5,7X,2PF6.2)	GUL	766
660 CONTINUE	GUL	767
IF(ICEK.EQ.1)ICEK=1	GUL	768
ERRSS=SQRT(SMERH/ICNT)	GUL	769
WRITE(6,662)ERRSS	GUL	770
662 FORMAT(//15X,18MHMS ERROR FOR CU =F8.4//)	GUL	771
WRITE(8) (CFPL(12),12=1,NOB),AK,ABC(3,1,1),ABC(4,1,1),MY,SAVE	GUL	772
REWIND 8	GUL	773
WRITE(6,66) (ABC(JVCX,1,1),JVCX=1,5)	GUL	774
66 FORMAT (//82M INTERPOLATIONS FOR THE FIELD OVER DIBEAM*OVRLAP	GUL	775
X HAVE JUST BEEN PERFORMED /54M BEAM DIA OVERLAP XPO	GUL	776
AS YPOS FIELD AVERAGE / 2X,50I2.5 //)	GUL	777
MY = .FALSE.	GUL	778
GO TO 999	GUL	779
.....	GUL	780
C INCREASE THE NUMBER OF GRID POINTS FOR COMPLEX FIELD	GUL	781
150 HEAD(IN,HEGRID)	GUL	782
NPSS = NPTS	GUL	783
NPYS = NPY	GUL	784
IF(INGRD.GT.NPTS)GO TO 151	GUL	785
GO TO 734	GUL	786
151 CALL HGRD(INGRD)	GUL	787
NOB = NPTS*NPY	GUL	788
WRITE(6,152)NPTS,NPYS,NPIS,NPY	GUL	789
152 FORMAT(//5X,21M YOUR ORIGINAL FIELD(,14,1M,13,39M) HAS BEEN REGR	GUL	790
1IDDED TO A LARGER SIZE (,14,1M,13,39M) TO GIVE THE FIELD MORE HOO	GUL	791
2M TO DO ITS THING//)	GUL	792
GO TO 999	GUL	793
.....	GUL	794
C RESONATOR CONVERGENCE TEST	GUL	795
70 NITEN = NITEN+1	GUL	796
WRITE(6,605) NITEN	GUL	797
605 FORMAT(////39M THIS IS THE COMPLETION OF ITERATION ,13 /)	GUL	798
IF(INIT.AND..NOT.HESTHT) GO TO 710	GUL	799
IF(.NOT.INIT) GO TO 720	GUL	800
GO TO 730	GUL	801
710 PCVNG=0.0	GUL	802
GO TO 720	GUL	803
730 HEAD(9) (CFIL(12),12=1,NOB)	GUL	804
REWIND 9	GUL	805
PCVNG=0.0	GUL	806
DO 740 IZ=1,NOB	GUL	807
PCVNG=PCVNG+CFIL(IZ)*CONJG(CFIL(IZ))	GUL	808

740	CONTINUE	GDL	809
	PCVNG=PCVNG*(X(2)-X(1))**2*(NPTS/NPY)	GDL	810
	IF(NNEG.EQ.1.OR.NNEG.EQ.2)PCVNG=PCVNG/WNU**2	GDL	811
720	FENN=1.00	GDL	812
	IF(PCVNG.GT.0.0)FENN=PPW/PCVNG-1.0	GDL	813
	IF(ABS(FENN).GT..007)ICEK=1	SOU77CY1	187
	PCVNGK=PCVNG/1000.	GDL	815
	WRITE(6,750)PPW,PCVNGK,FENN	GDL	816
750	FORMAT(30X,21M)FLUX CONVERGENCE TEST//10X,10MNEW FLUX =,0PG11.4.	GDL	817
	X12M OLD FLUX =,G11.4.4M ERROR =,F8.4////)	GDL	818
	PCVNG=PPW	GDL	819
	WRITE(9) (CU(IZ),IZ=1,NOB),X,UMX,DNY,NITEM	GDL	820
	HEWINDU 9	GDL	821
	IF(ICEK.EQ.0) GO TO 565	GDL	822
	IF(ICAV.GT.0) CALL NEGAIN(CT, NITEM)	GDL	823
	IF(NITEM.NE.1.GE.IGUL(99)) GO TO 1001	GDL	824
	READ(8) (CU(IZ),IZ=1,NOB),X,UMX,DNY,NITEM	GDL	825
	HEWINDU 8	GDL	826
	IF(ICAV.GT.0) GO TO 99	GDL	827
C	NORMALIZATION OF INPUT FIELD FOR BARE RESONATOR	GDL	828
	IF(.NOT.INIT) GO TO 86	GDL	829
	FMAX = 0.	GDL	830
	DO 87 IX=1,NOB	GDL	831
	IF(CABS(CU(IX)).LE.FMAX) GO TO 87	GDL	832
	FMAX = CABS(CU(IX))	GDL	833
	NPO1 = IX	GDL	834
87	CONTINUE	GDL	835
86	TEST=CABS(CU(NPO1))	GDL	836
	DO 77 IX=1,NOB	GDL	837
77	CU(IX) =CU(IX) /TEST	GDL	838
	GO TO 94	GDL	839
1001	HEAD(9) (CU(IZ),IZ=1,NOB),X,UMX,DNY	GDL	840
	HEWINDU 9	GDL	841
	GO TO 1000	GDL	842
C	*****	GDL	843
C	CALCULATE UCALC FLUX AND MINIMUM AND APERTURE LOSSES	GDL	844
949	PPW = 0.	GDL	845
	NOB=NPTS*NPY	GDL	846
	DO 78 IZ=1,NOB	GDL	847
78	PPW=PPW+CU(IZ)*CUNJG(CU(IZ))	GDL	848
	PPW=PPW*(X(2)-X(1))**2*(NPTS/NPY)	GDL	849
	IF(NNEG.EQ.1.OR.NNEG.EQ.2)PPW=PPW/WNU**2	GDL	850
	PBMIN=PPW	GDL	851
	GO TO (998,997,998,996,997),IGNAL	GDL	852
997	PBMIN=PPW/NMIRN	GDL	853
	PMINL=(PBMIN-PPW)/1000.	GDL	854
	PMINLP=(PBMIN-PPW)/PBMIN*100.	GDL	855
	WRITE(6,995)PMINL,PMINLP	GDL	856
995	FORMAT(17M)MINIMUM LOSS =,G12.4.1M=F8.2.8M PERCENT)	GDL	857
	IF(IGNAL.EQ.5)GO TO 998	GDL	858
996	APLUS=(SPPW-PBMIN)/1000.	GDL	859
	APLOSP=(SPPW-PBMIN)/SPPW*100.	GDL	860
	IF(ICNTL.EQ.1)GO TO 998	GDL	861
	WRITE(6,994)APLUS,APLOSP	GDL	862
994	FORMAT(17M)APERTURE LOSS =,G12.4.1M=F8.2.8M PERCENT)	GDL	863
998	PPW=PPW/1000.	GDL	864
	IGNAL=1	GDL	865
	SPPW=PPW	GDL	866
	UCALC=PX(NPTS)-2.*X(1)*X(2)	GDL	867
	IF(MSTEP.NE.1) WRITE(6,79)PPW,UCALC	GDL	868
79	FORMAT(///38M)ELEMENT TRANSMISSION FUNCTION APPLIED/8X,12MOCALC FL	GDL	869
	XUX =, G12.4/8X,12MOCALC =,F8.2)	GDL	870
	IF(MSTEP.EQ.1)WRITE(6,779)PPW	GDL	871
779	FORMAT(///38M)PROPAGATION STEP HAS BEEN APPLIED/ 8X,12MOCALC FL	GDL	872
	XUX =, G12.4)	GDL	873
	MSTEP=0	GDL	874
3623	IF(IPLOTS.EQ.0) GO TO 3624	GDL	875
	WRITE(6,3645) (NOT(ICNTL.1),I=1,20)	GDL	876
	CALL IPLUT(IPLOTS)	GDL	877
	IF(IGATE.NE.0) GO TO 3625	GDL	878
3645	FORMAT(25M) PLOTS AFTER STEP ***** ,2UA4. 6M*****	GDL	879
3624	IF(PPW.LE.0.)GO TO 732	GDL	880

3625 IF(.NOT.INIT)GO TO 98	GUL	881
GO TO 1000	GUL	882
505 WRITE(6,900)NITER	GUL	883
600 FORMAT(// 120(1H*))//49H	GUL	884
X H.14.14H ITERATIONS //120(1H*)//)	GUL	885
IF(KAUTO.EQ.1)GO TO 98	GUL	886
GO TO 1000	GUL	887
900 RETURN	GUL	888
732 WRITE(6,733)	GUL	889
733 FORMAT(//81H ALL RIGHT THEMES AIN T NO POWER IN THIS HERE BEAM A	GUL	890
AND THE REASON WE RE ALL HERE/50H IS POWER SO THIS JOB IS GOING	GUL	891
ATU LED AND KILLED QUICK /2JM ***CHECK INPUT** //)	GUL	892
STOP	GUL	893
734 WRITE(6,735)NGRD,NPTS	GUL	894
735 FORMAT(//5X.26H*X*X*X* VALUES OF NGRD (.1+.12H) AND NPTS (.1+.12H)	GUL	895
150H) MAKE THIS OPERATION UNNECESSARY UN WHUNG *X*X*X*//)	GUL	896
STOP	GUL	897
END	GUL	898

14. SUBROUTINE INTERP

a. Purpose -- Subroutine INTERP performs linear interpolation on two-dimensional real functions and on the real and imaginary parts of two-dimensional complex functions. Figure 32 describes the subroutine INTERP organization.

b. Relevant formalism -- Consider first the one-dimensional case in Figure 33. Assume the function value f is desired at a point x^* , between points x_1 and x_2 , with associated function values f_1 and f_2 , respectively:

Linear interpolation between f_1 and f_2 yields f as

$$f(x^*) \approx f_1 + \frac{(x - x_1)}{(x_2 - x_1)} (f_2 - f_1) \quad (108)$$

where the \approx is used since we are approximating f over the subinterval (x_1, x_2) .

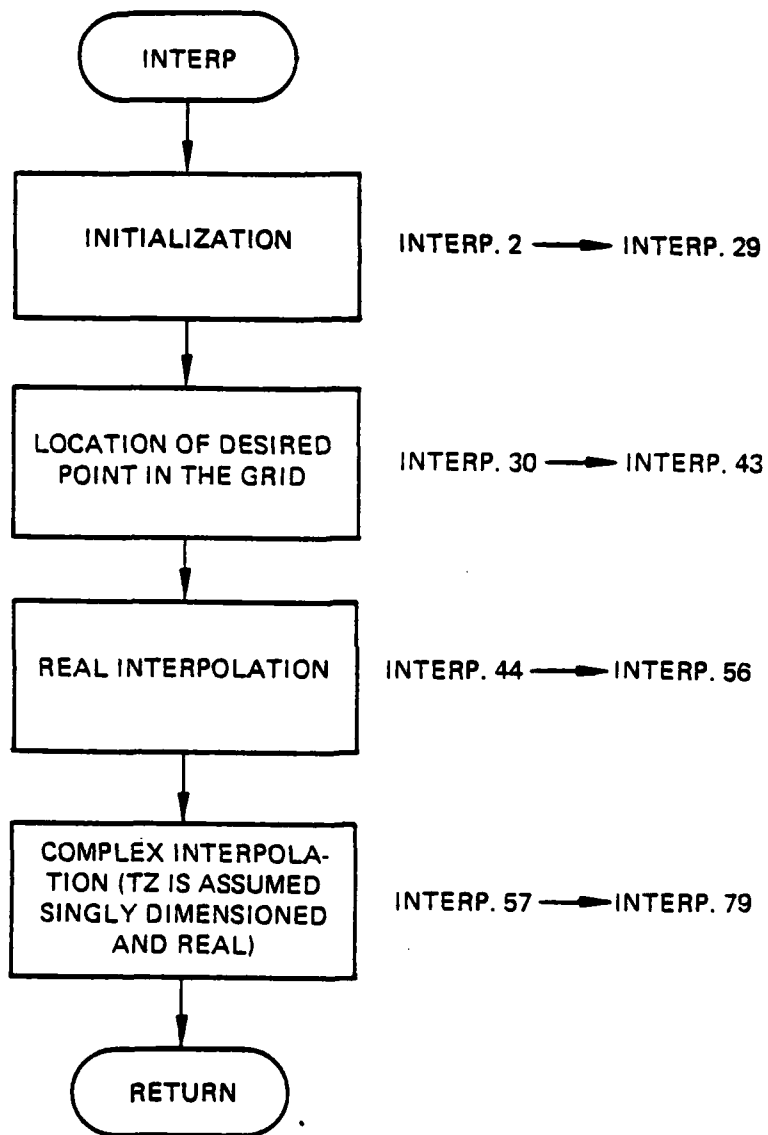


Figure 32. Subroutine INTERP organization.

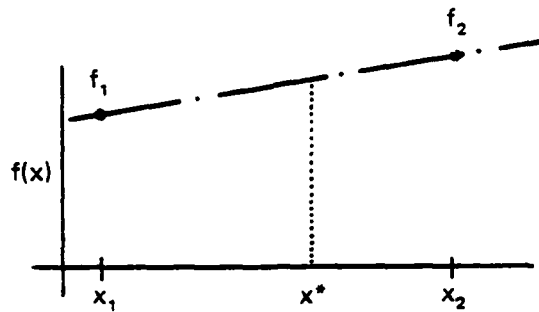


Figure 33. One-dimensional function case.

For the two dimensional case in Figure 34, subroutine INTERP establishes the location of the far corners of the rectangle bounding the desired point (x,y) , then linearly interpolates across top and bottom to find the two values at x . It then interpolates between these two points to find the value at (x, y) :

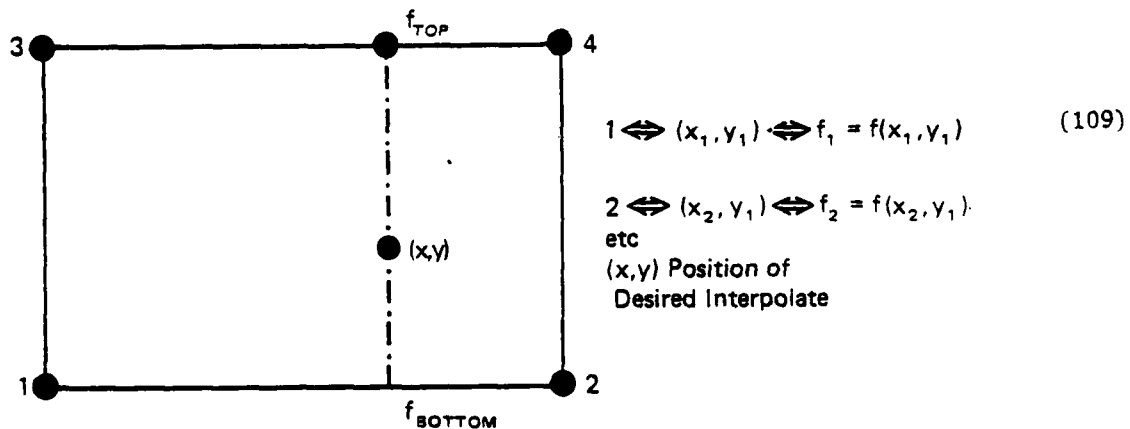


Figure 34. Two-dimensional function case.

$$f(x, y_2) \approx f_{\text{TOP}} = f_3 + \frac{(x - x_1)}{(x_2 - x_1)} (f_4 - f_3) \quad (110)$$

$$f(x, y_1) \approx f_{\text{BOTTOM}} = f_1 + \frac{(x - x_1)}{(x_2 - x_1)} (f_2 - f_1)$$

$$f(x,y) = f_{\text{BOTTOM}} + \left[\frac{(y - y_1)}{(y_2 - y_1)} \right] * (f_{\text{TOP}} - f_{\text{BOTTOM}})$$

c. Fortran

Arguments:

TXY = an array containing coordinate information
 (XIN, YIN) = the point at which the function value is desired
 TZ = the function to be interpolated
 TYPE = 1 real
 = 2 complex
 ZZ = two element array containing the interpolated value.

Note: If TZ is real, ZZ must still be dimensioned to 2 in the calling program, then the first element used as the answer.

NSYM = 1 symmetric,
 = 0 nonsymmetric

Note: Interpolation outside the region of definition of the distribution returns (0.0, 0.0) as the value of the interpolate.

There are no commons and no other subroutines are called.

Computer printouts of subroutine INTERP follow.

SUBROUTINE INTERP 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

SUBROUTINE INTERP(TAY,XIN,YIN,TZ,TYPE,ZZ,NSYM)	INTERP	2
C	INTERP	3
C THIS ROUTINE DOES A LINEAR INTERPOLATION ON THE	INTERP	4
C ARRAY TZ TO FIND THE VALUE ZZ AT XIN, YIN	INTERP	5
C THE (X,Y) GRID OF TZ IS CONTAINED IN THE ARRAY TAY	INTERP	6
C AS FOLLOWS:	INTERP	7
C TAY(1) = DX=SPACING BETWEEN X POINTS	INTERP	8
C TAY(2) = DY=SPACING BETWEEN Y POINTS	INTERP	9
C TAY(3) = NY, NO. OF POINTS ALONG Y-AXIS	INTERP	10
C TAY(4) = NX, NO. OF POINTS ALONG X-AXIS	INTERP	11
C TAY(5) = Y(1), MIN. Y VALUE	INTERP	12
C TAY(6+NY) = Y(NY), MAX. Y VALUE	INTERP	13
C TAY(7+NY) = X(1), MIN. X VALUE	INTERP	14
C TAY(8+NY+NX) = X(NX), MAX. X VALUE	INTERP	15

C	NO IS MAX. DIMENSION OF FIRST VARIABLE IN T2(I,J)	INTERR	16
C	TYPE = 1 T2 IS REAL ARRAY	INTERR	17
C	= 2 T2 IS COMPLEX ARRAY	INTERR	18
	LEVEL 2: T2,ZZ	INTERR	19
	DIMENSION ZZ(2),TX(1),TZ(1)	INTERR	20
	INTEGER TYPE, COMPLX	INTERR	21
	COMPLEX CZ2,CZ1,CZ2,CZ3,CZ4,CZA,CZB	INTERR	22
	DATA COMPLEX / 2 /	INTERR	23
	DX = TX(1)	INTERR	24
	OY = TX(2)	INTERR	25
	NY = TX(3)+.00001	INTERR	26
	NX = TX(4)+.00001	INTERR	27
	ZZ(1) = U.	INTERR	28
	ZZ(2) = U.	INTERR	29
C	TEST TO SEE IF XIN,YIN LIE WITHIN DEFINED T2 REGION	INTERR	30
	IF(XIN.LT.TX(5+NY)) GO TO 1000	INTERR	31
	IF(XIN.GT.TX(4+NX+NY)) GO TO 1000	INTERR	32
	IF(YIN.LT.TX(5)) GO TO 1000	INTERR	33
	IF(YIN.GT.0..AND.NSYM.EQ.1) GO TO 1000	INTERR	34
	IF(YIN.GT.TX(NY+4).AND.NSYM.EQ.0) GO TO 1000	INTERR	35
C	FIND POSITION OF (XIN,YIN) IN GRID	INTERR	36
	I1 = 1+(XIN-TX(5+NY))/DX	INTERR	37
	J1 = 1+(YIN-TX(5))/OY	INTERR	38
	IF(I1.EQ.NX) I1=I1-1	INTERR	39
	IF(J1.EQ.NY.AND.NSYM.EQ.0) J1=J1-1	INTERR	40
	SX = (XIN-TX(I1+4+NY))/DX	INTERR	41
	SY = (YIN-TX(J1+4))/OY	INTERR	42
C	FIND T2 VALUES AT I1,I1+1,J1,J1+1	INTERR	43
	IF(TYPE.EQ.COMPLX) GO TO 200	INTERR	44
C	T2 IS TREATED AS REAL ARRAY	INTERR	45
	IJ = I1+NX*(J1-1)	INTERR	46
	Z1 = TZ(IJ)	INTERR	47
	Z2 = TZ(IJ+1)	INTERR	48
	IJ = I1+NX*(J1)	INTERR	49
	IF (J1.EQ.NY) IJ=IJ-NX	INTERR	50
	Z3 = TZ(IJ)	INTERR	51
	Z4 = TZ(IJ+1)	INTERR	52
	ZA = Z1+5X*(Z2-Z1)	INTERR	53
	ZB = Z3+5X*(Z4-Z3)	INTERR	54
	ZZ(1) = ZA+SY*(ZB-ZA)	INTERR	55
	GO TO 1000	INTERR	56
	200 CONTINUE	INTERR	57
C	T2 IS TREATED AS COMPLEX ARRAY	INTERR	58
	IJ = TYPE*(I1+NX*(J1-1)) - 1	INTERR	59
	Z1A = TZ(IJ)	INTERR	60
	Z1B = TZ(IJ+1)	INTERR	61
	CZ1 = CMPLX(Z1A,Z1B)	INTERR	62
	Z2A = TZ(IJ+2)	INTERR	63
	Z2B = TZ(IJ+3)	INTERR	64
	CZ2 = CMPLX(Z2A,Z2B)	INTERR	65
	IJ = TYPE*(I1+NX*J1) - 1	INTERR	66
	IF (J1.EQ.NY) IJ=IJ-NX*TYPE	INTERR	67
	Z3A = TZ(IJ)	INTERR	68
	Z3B = TZ(IJ+1)	INTERR	69
	CZ3 = CMPLX(Z3A,Z3B)	INTERR	70
	Z4A = TZ(IJ+2)	INTERR	71
	Z4B = TZ(IJ+3)	INTERR	72
	CZ4 = CMPLX(Z4A,Z4B)	INTERR	73
	CZA = CZ1+5X*(CZ2-CZ1)	INTERR	74
	CZB = CZ3+5X*(CZ4-CZ3)	INTERR	75
	CZZ = CZA+SY*(CZB-CZA)	INTERR	76
	ZZ(1) = REAL(CZZ)	INTERR	77
	ZZ(2) = AIMAG(CZZ)	INTERR	78
	1000 RETURN	INTERR	79
	END	INTERR	80

15. SUBROUTINE IPLOT

a. Purpose -- Subroutine IPLOT has two major purposes: One is to create a printer iso-intensity plot. The other is to find the maximum intensity and to print the first title used by subroutine OUTPUT. It also contains the necessary information used by both subroutines OUTPUT and OUTPUR to determine whether a particular slice plot should be printed. Figure 35 describes the subroutine IPLOT organization.

b. Relevant formalism -- The output of this subroutine is an array of one-digit adjacent members with at least one asterisk, which indicates the maximum intensity points. The numbers indicate relative intensities.

c. Fortran

Argument List

The only argument of subroutine IPLOT is the parameter IPLTS which contains the information needed by OUTPUT (and OUTPUR) as well as IPLOT. IPLTS is filled with zero to five digits, each of which is 0 or 1. If it is 0, the indicated plot is not done; if 1, it is plotted. Assuming that the five digits of IPLTS are written ABCDE, the associated plots are:

- A: Radial (calls OUTPUR - not available)
- B: Iso-intensity
- C: X-axis slice plot
- D: Diagonal slice plot
- E: y-axis slice plot

Common Parameters:

The only common modified is CFIL due to its equivalence with US, the intensity array. The other parameters have then usual meaning including PLOTSG.

Recall: PLOTSG > 0 + intensity slice plots
 = 0 + no plots
 < 0 + amplitude slice plots

Subroutines called: OUTPUT, OUTPUR.

Computer printout of subroutine IPLOT follows.

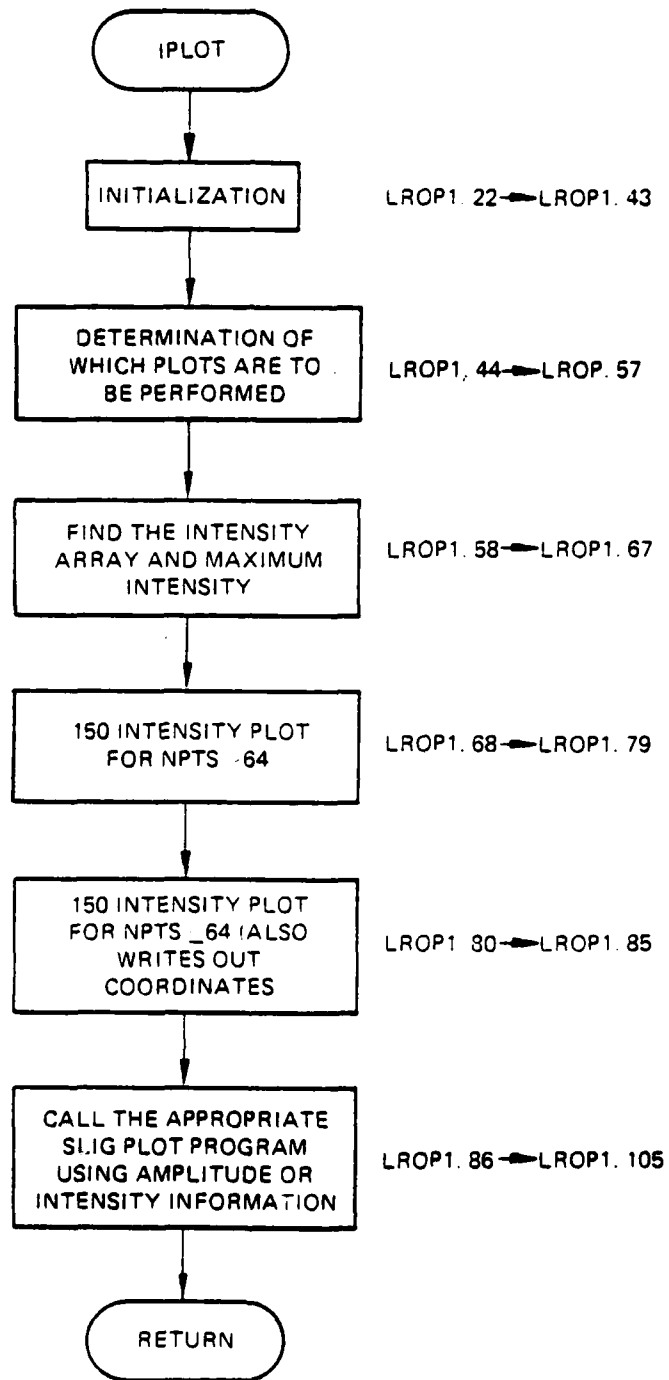


Figure 35. Subroutine IPLIT organization.

```

SUBROUTINE IPLOT(IPLTS)
C ISO-INTENSITY PLOT
C THIS ROUTINE MAKES A PLOT OF INTENSITY WHERE THE DIGIT
C PRINTED * IS DECILE OF PEAK INTENSITY FOR THAT ELEMENT.
LEVEL 2, CUR, US
COMMON/MLT/CUR(32768),CFIL(16512),X(128),NL,NPTS,NPY,UMX,UMY
COMMON/WAY/WNOW,NREG,NAPTH
COMMON /PLTSIG/ PLTSG
DIMENSION US(16384),II(150)
INTEGER II,BLANK,DUT
LOGICAL ISOIP,XAXIS,DIAG,YAXIS,HAUPLI
COMPLEX CFIL
EQUIVALENCE(CFIL(1),US(1))
DATA IBCUR / *M /, IBCUA / *M /
IF (PLTSG.EQ.0.) RETURN
IMHX=IBCDX
HAUPLI=.FALSE.
ISOIP=.FALSE.
XAXIS=.FALSE.
DIAG=.FALSE.
YAXIS=.FALSE.
IPL=IPLTS
IF (IPL.LT.10000) GO TO 290
RAUPLI = .TRUE.
IPL = IPL - 10000
IMHX = IBCDX
290 IF (IPL.LT.1000) GO TO 300
ISOIP=.TRUE.
IPL = IPL - 1000
300 IF (IPL.LT.100) GO TO 400
XAXIS = .TRUE.
IPL = IPL - 100
400 IF (IPL.LT.10) GO TO 500
DIAG = .TRUE.
IPL = IPL - 10
500 IF (IPL.NE.0) YAXIS=.TRUE.
PI=3.141592
OX=X(2)-X(1)
XDIM=OX*NPTS
NOB=NPTS*NPY
XFACT=1.
IF (NREG.EQ.1.OR.NREG.EQ.2) XFACT=1./WNW**2
UMAX=0.
DO 1 J=1,NOB
US(J) = (CUR(2*J-1)**2 + CUR(2*J)**2) * XFACT
1 UMAX=AMAX1(UMAX,US(J))
IF (.NOT.ISOIP) GO TO 98
UMAX=UMAX/1000.
IF (NPY.LE.64) WRITE(6,5) IMHX
5 FORMAT(1X,A1)
IF (NPY.LE.64) GO TO 99
DO 4 J=1,NPTS
DO 2 I=1,NPY
I2 = J * (I-1)*NPTS
2 II(I)=10.*US(I2)/UMAX
4 WRITE(6,3) (II(I),I=1,NPY)
3 FORMAT(1X,12B11)
GO TO 98
99 DO 14 J=1,NPTS
DO 12 I=1,NPY
I2 = J * (I-1)*NPTS
12 II(I)=10.*US(I2)/UMAX
14 WRITE(6,13) X(J),(II(I),I=1,NPY)
13 FORMAT(1X,F10.2,2X,64I1)
98 WRITE(6,6) XDIM,UMAX,UMX,UMY
6 FORMAT(1UMD UCALC = ,G11.5,4X,7MIMAX = ,G11.4//24X,
X 39M THE CENTER OF THE BEAM IS LOCATED AT (,F6.3,1M,,F6.3,1M))

```

IF (PLOTSG.GT.0.) GO TO 1500	LHUP1	89
IF (.NOT.RADPLT) WRITE (6,7)	LHUP1	90
7 FORMAT (LHUP1	91
X90)AMPLITUDE, PHASE PLOTTED IN THE X-DIRECTION THROUGH THE CENTE	LHUP1	92
XR OF DCALC (J=NPTS/2)	LHUP1	93
UMAXA=SQRT(UMAX)	LHUP1	94
GO TO 1550	LHUP1	95
1500 WRITE (6,786)	LHUP1	96
786 FORMAT (LHUP1	97
X90)INTENSITY, PHASE PLOTTED IN THE X-DIRECTION THROUGH THE CENTE	LHUP1	98
XR OF DCALC (J=NPTS/2)	LHUP1	99
UMAXA = UMAX	LHUP1	100
1550 IF (NREG.NE.0.AND.PLOTSG.LT.0.) UMAXA=UMAXA*WNOW	LHUP1	101
IF (NREG.NE.0.AND.PLOTSG.GT.0.) UMAXA=UMAXA*WNOW**2	LHUP1	102
IF (.NOT.RADPLT) CALL OUTPUT(CUR,NPY,NPTS,X,J,UMAXA,XXIS,DIAG,	LHUP1	103
X YAXIS)	LHUP1	104
IF (RADPLT) CALL OUTPUT(CUR,NPY,NPTS,X,UMAXA,XXIS,DIAG,YAXIS)	LHUP1	105
RETURN	LHUP1	106
END	LHUP1	107

16. SUBROUTINE KINET

a. Purpose -- This subroutine calculates the kinetics and loaded gain in the gas dynamic laser cavity. It is called by GAINXY for either small signal gain calculation (along a single stream tube in the x-direction) or full field loaded gain along several stream tubes. Figure 36 describes the subroutine KINET flow chart.

An intensity field VIC and previous gain field GAN are brought in from GAINXY and are updated by recomputing the kinetics and gain in the cavity as a function of these updated fields. The population rate equations (i.e., the equations showing the rate at which the energy of each vibrational level is changing) are numerically integrated along the x(flow)-direction. This is continued along the x-direction until the end of the calculation region (IXMAX) and is then redone for each stream tube in the y-direction (if full loaded gain is requested by IFIELD \neq 1). The full gain field GAN (I) is then updated.

The assumption is made that the flow area of the cavity is constant through the region of interest for all kinetics calculations.

b. Relevant formalism -- Gain is calculated in the x-direction from nozzle exit plane to the end of the region of interest IXMAX at a constant y value, as shown in Figure 37. This is done along only one mid-cavity stream tube for small signal gain calculation and at every y-value (IY) for the full field loaded gain calculations.

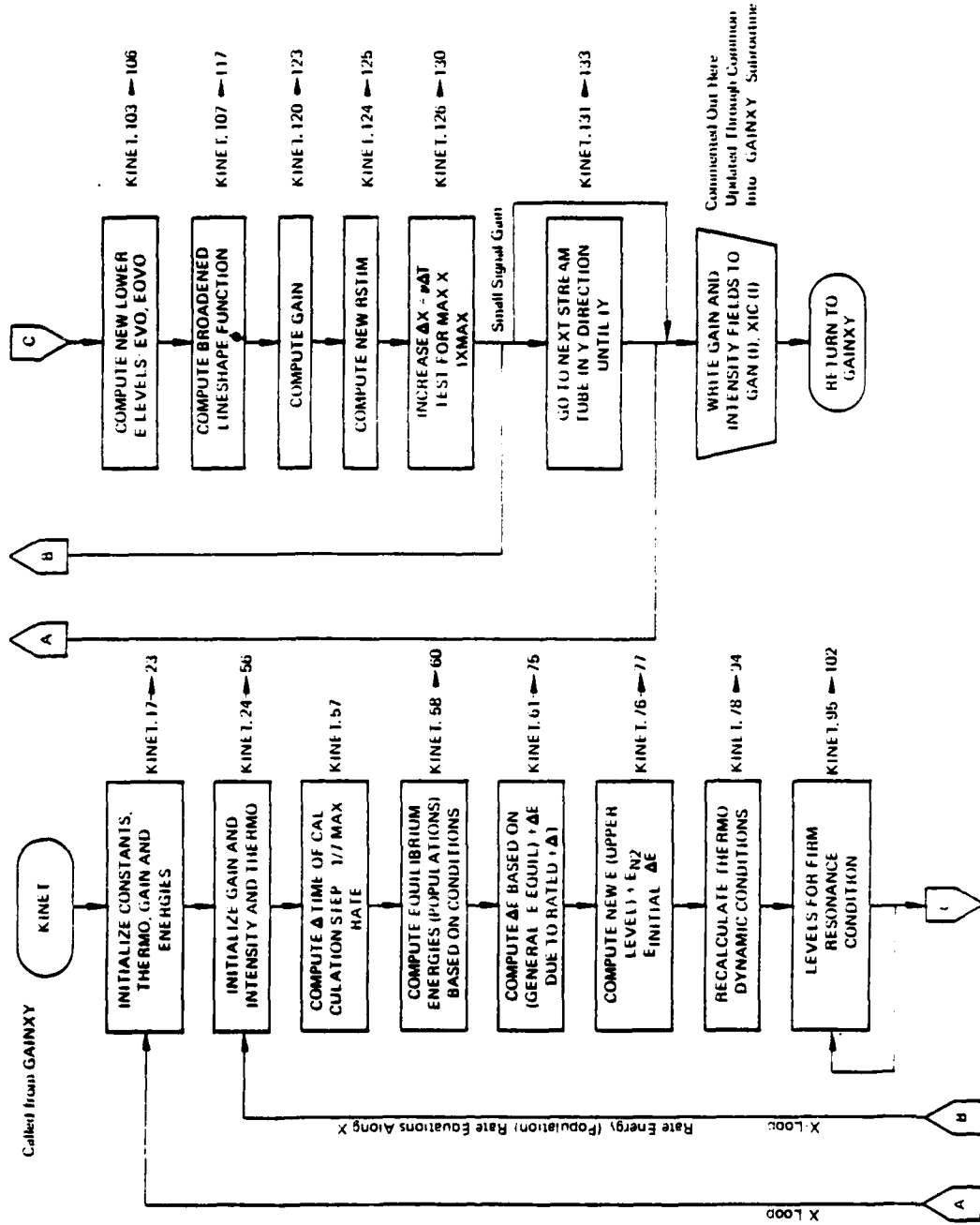


Figure 36. Subroutine KINET flow chart.

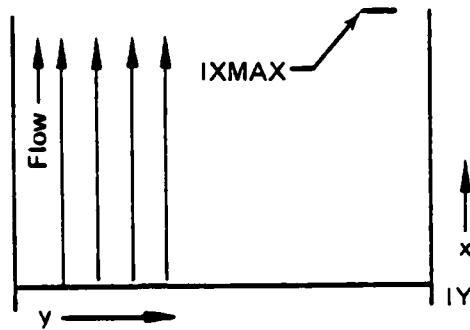


Figure 37. Region of interest IXMAX.

Rate equations are set up for each level which describe the energy in that level.

$$\frac{dE}{dt} \text{ upper} = \left[\frac{dE}{dt} \right]_{V-T} + \left[\frac{dE}{dt} \right]_{V-V} + \left[\frac{dE}{dt} \right]_{S.E.} \quad (111)$$

$$\frac{dE}{dt} \text{ lower} = \left[\frac{dE}{dt} \right]_{V-T} + \left[\frac{dE}{dt} \right]_{V-V} + \left[\frac{dE}{dt} \right]_{S.E.} \quad (112)$$

$$\frac{dE}{dt} N_2 = \left[\frac{dE}{dt} \right]_{V-T} + \left[\frac{dE}{dt} \right]_{V-V} \quad (113)$$

The energies of each level EN2, EOOV, EVOO and EOVO are updated at each ΔX step, i.e., the ΔE change is computed and the corresponding heat addition (local temperature change) is used to compute the energy in the subsequent step.

The stimulated emission energy rate can be used to determine local intensity change and, hence, gain. Energies of levels are described by population densities n_u and n_L :

$$\frac{dI_\nu}{ds} \Big|_{u \neq L} = h\nu \left\{ n_u \bar{N}_{UL}(\nu) A_{UL}(\nu) - \left[n_L \bar{N}_{LU}(\nu) B_{LU} - n_u \bar{N}_{UL}(\nu) B_{LU} \right] I_\nu \right\} \quad (114)$$

where I_ν is the specific intensity at the frequency ν ; n_U is the population density of the upper level; n_L , that of the lower level; A_{UL} the Einstein coefficient for spontaneous emission; B_{UL} , the stimulated emission coefficient; and B_{LU} , for absorption. The quantities η_{UL} , N_{UL} and ϕ_{LU} are the line shape functions for the three respective processes, which are generally different.

Characteristic times for the spontaneous decay of low-lying vibrational states for molecular species of interest are of the order 10^{-1} to 10^{-5} second, whereas other rate processes are typically much faster. Hence, in the equation above, the spontaneous emission term generally can be neglected. Also, for the present analyses, interest focuses primarily on photon processes occurring at line center. At line center $\phi_{LU} = \eta_{UL}$. Thus,

$$\frac{dI_\nu}{ds} = h\nu\phi_{LU}(\nu_\sigma) \left[B_{UL} n_u - B_{LU} n_L \right] I_{\nu_0} \quad (115)$$

The factor multiplying I_{ν_0} is the optical gain coefficient, viz:

$$g_{UL} = h\nu\phi_{LU} \left[B_{UL} n_u - B_{LU} n_L \right] \quad (116)$$

The Einstein coefficients are connected by the relationship

$$\frac{B_{LU}}{B_{UL}} = \frac{d_u}{d_L} \quad (117)$$

where d_U and d_L are degeneracies (statistical weights) of the upper and lower states, respectively. Also, it is possible to write

$$B_{LU} = \frac{8\pi^3}{3h^2c} \left| R_{LU} \right|^2 \quad (118)$$

where R_{LU} is the quantum-mechanically-derived transition matrix element. Hence, the gain expression may be rewritten as

$$g_{UL} = \frac{8\pi^3}{3h} \left(\frac{\nu_0}{c}\right) \phi_{LU}(\nu_0) |R_{LU}|^2 \left[\frac{n_u}{d_u} - \frac{n_L}{d_L} \right] \quad (119)$$

or

$$g_{v_j}^{j'} = \frac{8\pi^3}{3h} \frac{\nu_0}{c} \phi_{LU} \nu_0 |R_{LU}|^2 \left[\frac{n_{v_j}}{d_{v_j}} - \frac{n_{v_j'}}{d_{v_j'}} \right] \quad (120)$$

Consider vibrational-rotational transitions of the form

$$(v+1, J) \leftarrow (v, J)$$

where v is the vibrational quantum number and J is the rotational quantum number.

Then

$$|R_{LU}|^2 = S_j |R_{v, v+1}|^2 \quad (121)$$

where:

$$S_j = \begin{cases} J & \text{for P-branch transitions (i.e., } J' = J + 1) \\ J + 1 & \text{for R-branch transitions (i.e., } J' = J - 1) \end{cases}$$

$$R_{v, v+1} = \text{vibrational-transition matrix element}$$

At pressures of a few torr or less, transitions are predominately Doppler broadened. At higher pressures, the combined influence of Doppler and pressure (Lorentz) broadening is present. Therefore, the line-shape factor $\phi_{LU}(\nu_0)$ is represented in terms of a Voigt profile such that

$$\frac{v_0}{c} \rho_{Lu}(v_0) = \left(\frac{m}{2\pi kT} \right)^{1/2} \exp(-\xi)^2 \operatorname{erfc}(\xi) = \left(\frac{m}{2\pi kT} \right)^{1/2} \phi(\xi) \quad (122)$$

$$a_D(v_0) = \left(\frac{K}{n} \right) \left(\begin{array}{c} \theta_{001} - \theta_{001} + J(J+1) \theta_{\text{rot}}^{001} - J'(J'+1) \theta_{\text{rot}}^{100} \\ (\theta_{02^0 0}) \qquad \qquad \qquad (\theta_{\text{rot}}^{020}) \end{array} \right) \quad (123)$$

$$\sigma_{\text{CO}_2 - \text{CO}_2} = \frac{001 \rightarrow 100}{10.5 \times 10^{-15} \text{ cm}^2} \quad (124)$$

$$\sigma_{\text{CO}_2 - \text{CO}_2} = \frac{001 \rightarrow 02^0 0}{10.2 \times 10^{-15} \text{ cm}^2} \quad (125)$$

The influences of Doppler broadening and vibration-rotation interaction have been taken into account.

where

$$\xi = \frac{a_p}{a_D} \sqrt{\ln 2} \quad (126)$$

a_p = pressure broadened (Lorentz) half-width

$$= \frac{n}{2\pi c} \sum_s v_s X_s \sigma_s$$

a_D = Doppler broadened half-width

$$= \frac{v_0}{c} \sqrt{\frac{2kT(\ln 2)}{m}}$$

v_s is the mean relative velocity ($\sqrt{2kT/M}$) between the emitting molecule and the colliding species; X_s is the species mole fraction, σ_s is the broadening cross-section due to the impacting species s ; v_0 is the transition frequency at line center; m is the mass of the emitter molecule; and M is the reduced mass between an emitter molecule and the collider molecule of species s :

$$\mu = \frac{m m_s}{m + m_s} \quad (127)$$

The optical gain coefficient may be rewritten as

$$g(V, J) = \frac{8\pi^3}{3h} \left(\frac{m}{2kT} \right)^{1/2} \phi(\xi) S_J \left| R_{V, v+1} \right|^2 \left[\frac{n_{V+1, J'}}{d_J} - \frac{n_{V, J}}{d_J} \right] \quad (128)$$

Here the quantities V and J in the expression $g(V, J)$ indicate the lower levels of the transition.

In treating the populations of the vibrational-rotational levels, it is assumed that the rotational mode can be described by the local translational temperature T . Hence,

$$n_{V, J} = \left(\frac{n_{V, J}}{n_V} \right) n_V = \frac{d_J \exp \left[-1.439 J(J+1) (B_e - \alpha_e (v+1/2)/T) \right] n_V}{Q_{\text{rot}}^{(v)}} \quad (129)$$

where B_e is the spectroscopic rotational constant (cm^{-1}), and α_e is its anharmonic correction. The quantity $Q_{\text{rot}}^{(v)}$ is the rotational partition function, which is evaluated according to the relation

$$Q_{\text{rot}}^{(v)} = \sum_J (2J+1) \exp (-E_{\text{rot}}(J, V)/kT) \quad (130)$$

The populations can also be represented by:

$$n_{V, J} = n_V f_J = n_V \left[\frac{2J+1}{Q_{\text{rot}}^{(v)}} \right] \exp \left(\frac{-J(J+1)}{kT} \right) \vartheta_{\text{rot}}^{(v)} \quad (131)$$

where,

$$Q_{\text{rot}}^{(V)} = \frac{T}{2\theta_{\text{rot}}^{(V)}}$$

$$\frac{n_{VJ}}{g_{VJ}} = \frac{n_V}{g_V} \exp\left(\frac{-J(J+1)}{kT}\right) \theta_{\text{rot}}^{(V)}$$

$$\frac{n_V}{g_V} = n_{000} \exp(-\theta_V/T_V)$$

θ_V = characteristic temp. of state

T_V = vibrational temperature of state

g_V, g_{VS} represent degeneracies

For the transitions



the pertinent constants are:

$$R_{001,100} = 0.0331 \times 10^{-18} \text{ esu-cm}$$

$$R_{001,02^0 0} = 0.0295 \times 10^{-18} \text{ esu-cm}$$

$$\theta_{\text{rot}}^{(001)} = 0.55632 \text{ K}$$

$$\theta_{\text{rot}}^{(02^0 0)} = 0.56106 \text{ K}$$

$$\theta_{\text{rot}}^{(100)} = 0.56078 \text{ K}$$

$$\theta_{001} = 3380 \text{ K} \quad e_{100} = 1997 \text{ K}$$

$$\theta_{020} = 1850 \text{ K}$$

The expressions for the gain coefficients on two transitions are

$$g_{001,J}^{700,J} = (0.79 \times 10^{-14}) |m| (1 - 0.0044m) T^{-\frac{3}{2}} n X_{000} \phi \left[(0.55632) \exp\left(\frac{-3380}{T_{001}} - J(J+1) (0.55632/T)\right) - (0.56078) \exp\left(\frac{-2000}{T_{100}} - J'(J'+1) (0.56078/T)\right) \right] \quad (132)$$

$$g_{001,J}^{020,J'} = (0.63 \times 10^{-14}) |m| (1 - 0.006m) T^{-\frac{3}{2}} n X_{000} \phi \left[(0.55632) \exp\left(\frac{-3380}{T_{001}} - J(J+1) (0.55632/T)\right) - (0.56106) \exp\left(\frac{-1850}{T_{020}} - J'(J'+1) (0.56106/T)\right) \right] \quad (135)$$

where $m = -(j + 1)$ $J' = J + 1$ (P)

$m = J$ $J' = J - 1$ (R)

$n = \text{total number density} = \frac{P}{KT}$

$X_{000} = \text{mole fraction of ground state CO}_2 \text{ (from program)}$

$J' = 0, 2, 4, 6, \dots$

For largely pressure-broadened line, ϕ may be expressed as:

$$\phi \approx \frac{1}{\sqrt{\pi} \xi} \left[1 - \frac{0.5}{\xi^2} + \frac{0.75}{\xi^4} - \frac{1.875}{\xi^6} + \frac{6.5625}{\xi^8} - \dots \right] \quad (134)$$

Argument List

XIC	The field (matrix) of individual intensities in the calculation region
GAN	Gain (updated) of each of the point locations of the field
IXMAX	Number of points in the flow direction

DXCAV The distance between points in the x-direction
 IFIELD Indicator for small signal gain (IFIELD = 1) or Loaded
 Gain (IFIELD ≠ 1)
 IY Number of flow streams, i.e., points in the y-dimension.

Commons Modified

/PROPT/
 TS Static temperature in the cavity (K)
 PS Static pressure in the cavity
 V Gas velocity (cm/sec)
 RHO Gas density (gm/cc)
 RHON Number density (particles/cc)
 /ENERG/
 EN2 Energy (population) of the V = 1 level of N₂
 EOOV Energy (population) of the asymmetric stretch vibration mode
 EOVO Energy (population) of the bending vibration mode of CO₂
 EVOO Energy (population) of the symmetric stretch mode of CO₂
 /RATE/
 RSTIM Rate for stimulated emission.

SUBROUTINE KINET 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

	SUBROUTINE KINET(XIC,GAN,IAMAX,DXCAV,IFIELD,IY)	KINET	2
C	CO2 KINETICS ROUTINE	KINET	3
C	THIS ROUTINE CALCULATES GUL GAIN (10.0) AS A FUNCTION OF KINETIC	KINET	4
C	AND STIMULATED EMISSION EFFECTS.	KINET	5
	LEVEL 2, XIC,GAN	KINET	6
	COMMON/PROPT/TS,PS,V,HMO,HMUN,CP,GAMMA,H,B,XLAMB,MNU,CPHM	KINET	7
	COMMON/START/TSI,PSI,VI,EUVI,EUVUI,EVOUI,EN2I,GAINI	KINET	8
	COMMON/MULES/XN2,ACU2,AM2U,ACU,XU2	KINET	9
	COMMON/ENERG/EN2,EUVU,EUVUUI,EVOU	KINET	10
	COMMON/RATE/HN2,HCJ,HC2,HMUM, RSTIM	KINET	11
	COMMON/FACTER/XMU,AG,GCUN,MUTUP,MUTLU,MCUN,C	KINET	12
	DIMENSION GAN(1),XIC(1),SUEV(10)	KINET	13
	IF(IFIELD .EQ. 1) IY=1	KINET	14
C	IF(IFIELD .EQ. 1) CALL ZENU(XIC(1),XIC(1038))	KINET	15
	IF (IFIELD.NE.1) GO TO 174	KINET	16
	DO 173 IZENU=1,10384	KINET	17
173	XIC(IZENU) = 0.	KINET	18
174	F3 = 2.349E10/MNU	KINET	19
	F4 = 1.388E10/MNU	KINET	20
	F5 = GAMMA*H	KINET	21
	F6 = XMU/AG	KINET	22
	F7 = XCU2*2349.	KINET	23
	DO 200 J=1,17	KINET	24
	TS = TSI	KINET	25
	PS = PSI	KINET	26
	V = VI	KINET	27
	GAIN = GAINI	KINET	28
	HMO = PS/H/TS*1.013E6	KINET	29
	RHON = HMO/XMU*AG	KINET	30
	T2 = 459.8 / ALU(1.+XCU2*133./EUVUI)	SOU77CY1	188

EGL = EUV01 * EV001	KINET	32
EN2 = EN2I	KINET	33
EV00 = EV00I	KINET	34
EUV0 = EUV0I	KINET	35
EU0V = EU0VI	KINET	36
X = 0.0	KINET	37
SUMDEV = 0.0	KINET	38
IBAR = 0	KINET	39
XCAV = 0.0	KINET	40
10 IBAR = IBAR*1	KINET	41
XCAV = UACAV*(IBAR-1)+UACAV/2.	KINET	42
IF(XCAV.LT.X) GO TO 100	KINET	43
CALL MIX	KINET	44
20 G1 = GAIN	KINET	45
F1 = EXP(3354./TS)	KINET	46
F2 = EXP(3380./TS)	KINET	47
IF(IBAR.EQ.1) GO TO 6	KINET	48
IJ = (X+XCAV/2.)/UACAV	KINET	49
IP = IJ*(IJ-1)*IXMAX	KINET	50
XI = XIC(IP)*(XIC(IP+1)-XIC(IP))/UACAV*(X-IJ+XCAV+(UACAV/2.))	KINET	51
X)	KINET	52
GO TO 7	KINET	53
6 XI = XIC(1+(IJ-1)*IXMAX)*X/(UACAV/2.)	KINET	54
7 CONTINUE	KINET	55
SUM1 = SUMDEV	KINET	56
DT = 1./(1.0*AMAX1(NCZ,NPUMP,NSTIM))	KINET	57
EUN2 = XN2/(F1 -1.)*2331.	KINET	58
EU00V = XCU2*2349./(F2 -1.)	KINET	59
EU0VU = XCU2*1334./ (EXP(954.8/TS)-1.)	KINET	60
XA = 1.-EN2/EUN2	KINET	61
XB = 1.-EU0V/EU00V	KINET	62
EMSL = -25.4/TS	KINET	63
YA = 1.-1./F1	KINET	64
YB = 1.-1./F2	KINET	65
XAB = 1./YA*(XA-XB-(EMSL*XA*(XB-1.)))/(F1 -1.)	KINET	66
XADDT = -YA*XC02*XB*NPUMP	KINET	67
XADDT = YB*AN2*EXP(-EMSL)*XB*NPUMP	KINET	68
DEN2MP = (EN2-EUN2)*HN2*DT	KINET	69
DEN2 = EGN2*XADDT*DT + DEN2MP	KINET	70
F1U = XI*GAIN/HMON	KINET	71
DEU0VH = (EU0V-EU00V)*RCJ*DT	KINET	72
DEU0V = DEU0VH + (F3*F1U-EU00V*XADDT)*DT	KINET	73
DEU0U = (EU0U-EU00U)*RC2*DT	KINET	74
UEGL = DEU0U-1.094*DEN2MP-1.086*DEU0VH-F4*F1U*DT	KINET	75
EN2 = EN2-DEN2	KINET	76
EU0V = EU0V-UEGL	KINET	77
EGL = EGL-UEGL	KINET	78
SUMDEV = SUMDEV + DEU0U*V*1.987E-16*HMON	KINET	79
DX = V*DT	KINET	80
X = X + DX	KINET	81
PS = PS*1.013E6	KINET	82
DEV = DEU0U/DT*1.1967/E8	KINET	83
Q = V*V/(F5 *TS)-1.0	KINET	84
PH = DEV/CP/TS	KINET	85
V = V-PP/Q*V*DT	KINET	86
RMU = HMU*PH/Q*HMO*DT	KINET	87
HMON = HMO/F6	KINET	88
PS = PS*PS*PP*GAMMA*(Q+1.)/Q*DT	KINET	89
TS = PS/HMU/H	KINET	90
PS = PS/1.013E6	KINET	91
CH12 = -954.8/TS	KINET	92
Y = CH12	KINET	93
Z1 = EXP(77./TS)	KINET	94
31 F8 = EXP(-Y)	KINET	95
F9 = EXP(-2.*Y+77./TS)	KINET	96
FA = EGL-XC02*(1388./(F8*F9*Z1-1.))+1334./(F8-1.)	KINET	97
FP1 = XCU2*(2776.*F9/(F9-1.))*2*1334.*F8/(F8-1.)*2)	KINET	98
FMA = -FP1	KINET	99
YULD = Y	KINET	100
Y = YULD - FA/FMA	KINET	101
IF (ABS(Y-YULD) / Y).GT. 1.E-3) GO TO 31	KINET	102
T2 = -954.8 / Y	KINET	103

T1 = 1388./ (1334./T2 * 54./TS)	KINET	104
EV00 = 1388.*XC02/(EXP(1997./T1) - 1.)	KINET	105
EU00 = XC02*1334./ (EXP(959.8 /T2)-1.)	KINET	106
CH12 = Y	KINET	107
CH11 = 2. * CH12 - 77.71 / TS	KINET	108
Q1 = 1./ (1.-EXP(CH11))	KINET	109
Q2 = 1./ (1.-EXP(CH12))	KINET	110
Q3 = EU00/F7*1.	KINET	111
T3 = -3380./ALOG(1.-1./Q3)	KINET	112
X000 = XC02/(Q1*Q2*Q3)	KINET	113
APAD = CPHM*HMM	KINET	114
HMM = .8326*APAD	KINET	115
IF (W0HM.GT.10.) GO TO 40	KINET	116
PHI = EXP(W0HM*2)*ENFC(W0HM)	KINET	117
GO TO 41	KINET	118
40 PHI = 0.67764/APAD	KINET	119
41 CONTINUE	KINET	120
TFAC = TS*(1.5)	KINET	121
GAIN = GCON*TFAC*HMM*X000*PHI*(.556*EXP(-3380./T3-NUTUP/TS)	KINET	122
X -.561*EXP(-1997./T1-NUTLU/TS)	KINET	123
HIGSIG = GCON*TFAC*PHI*EXP(-NUTUP/TS)*.556	KINET	124
HSTM = XI*HIGSIG/HMU*1.E7	KINET	125
IF (X.LE.XCAV) GO TO 20	KINET	126
100 GAN(I*HMM*(J-1)*IXMAX) = GAIN-(GAIN-G1)*(X-XCAV)/DX	KINET	127
SDEV(I*HMM) = SUMDEV-(SUM1-SUMDEV)*(X-XCAV)/DX	KINET	128
IF (I*HMM.EQ.IXMAX) GO TO 300	KINET	129
GO TO 10	KINET	130
300 DO 301 I = 1,IXMAX	KINET	131
301 XIC(I*(J-1)*IXMAX) = SDEV(I)	KINET	132
200 CONTINUE	KINET	133
DO 60 J=1,IY	KINET	134
C WRITE(6,205) (XIC(I,J),I=1,IXMAX)	KINET	135
C WRITE(6,203) (GAN(I,J),I=1,IXMAX)	KINET	136
C 60 WRITE(6,204) (SDEV(I),I=1,IXMAX)	KINET	137
C WRITE(6,201) X,EN2,EU0V,EGL	KINET	138
C WRITE(6,202) TS,PS,V,HMU,D	KINET	139
C 201 FORMAT(5X,24H--KINET-- X,EN2,EU0V,EGL,5X,4E15.5/)	KINET	140
C 202 FORMAT(10X,24H--KINET-- TS,PS,V,HMU,D,5X,5E15.5/)	KINET	141
C 203 FORMAT(10X,13H--KINET-- GAN/5(10E12.4/))	KINET	142
C 204 FORMAT(10X,14H--KINET-- SDEV/5(10E12.4/))	KINET	143
C 205 FORMAT(10X,19H--FIELD INTENSITY--/5(10E12.4/))	KINET	144
RETURN	KINET	145
END	KINET	146

17. SUBROUTINE MIRROR

a. Purpose -- MIRROR applies a mirror transmission function to the complex field which may include reflectivity, clipping, radius of curvature, edge diffraction imaging, small tilt, astigmatism, localized surface distortion, and overall spherical distortion. In addition, two specialized options have been included: (1) a toric mirror effect for axicon optics and (2) a mirror dimple effect which enables a localized difference in radius of curvature. Figure 38 shows the subroutine MIRROR organization. Computer printouts of the MIRROR subroutine begin on page 168.

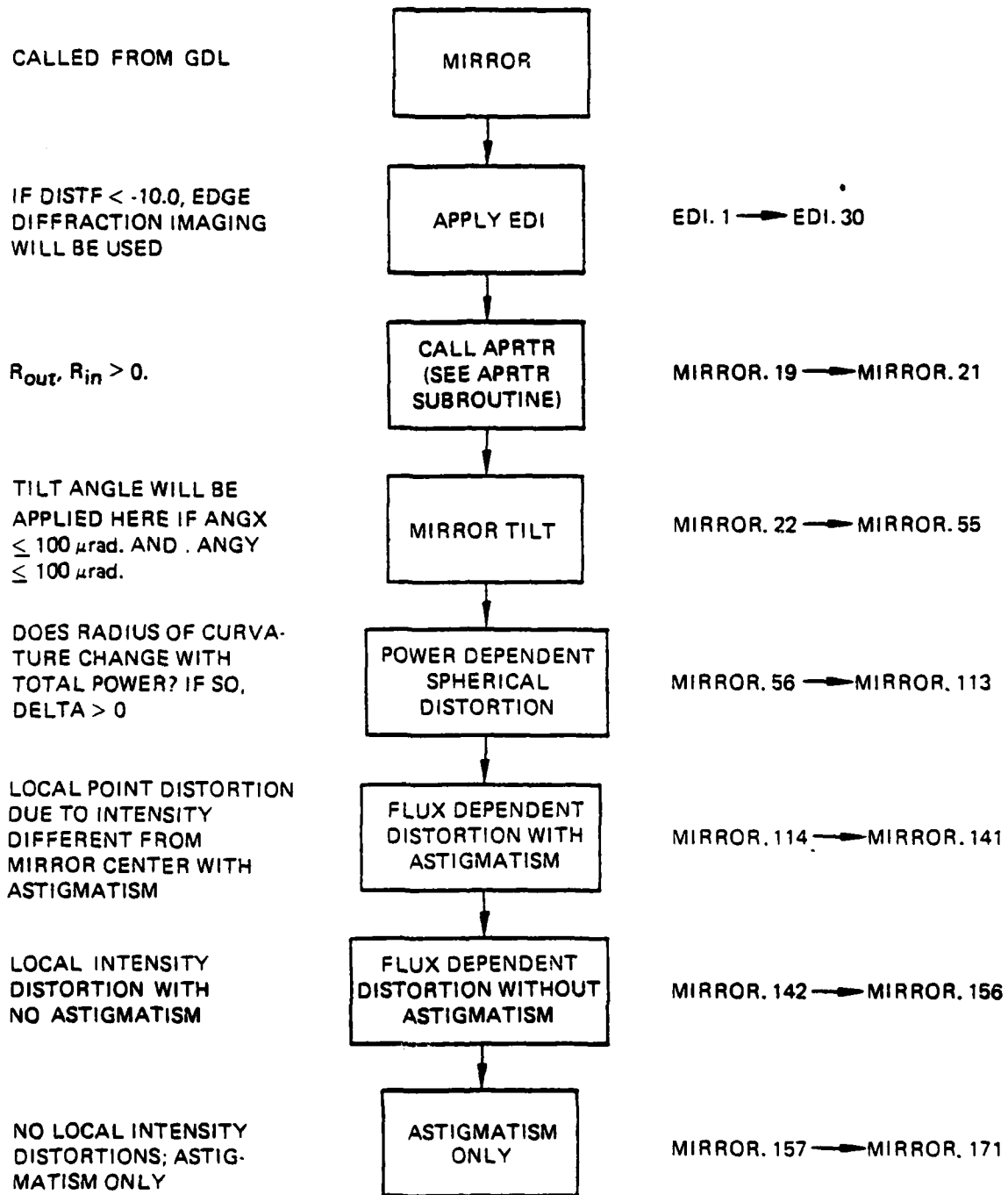


Figure 38. Subroutine MIRROR organization.

The routine first tests for the option of edge diffraction imaging in which the outer annular edge of the mirror has a radius of curvature different from the mirror. When this option is used the MIRROR subroutine must be called separately to apply EDI.

The subroutine must be called again for the rest of the mirror.

The routine then apertures the field to the size of the mirror and applies small mirror misalignments (angles less than 100 microradians) to the field. For large angles, the angle information is stored in ANGX and ANGY which are located in common MRPROP and used to later determine the location of the center of the field. The field itself is not altered for the large angles.

b. Relevant formalism -- A distortion-free mirror is applied to the field in Figure 39 by changing the optical path lengths of the field points. For example, apply a convex mirror to a plane wave.

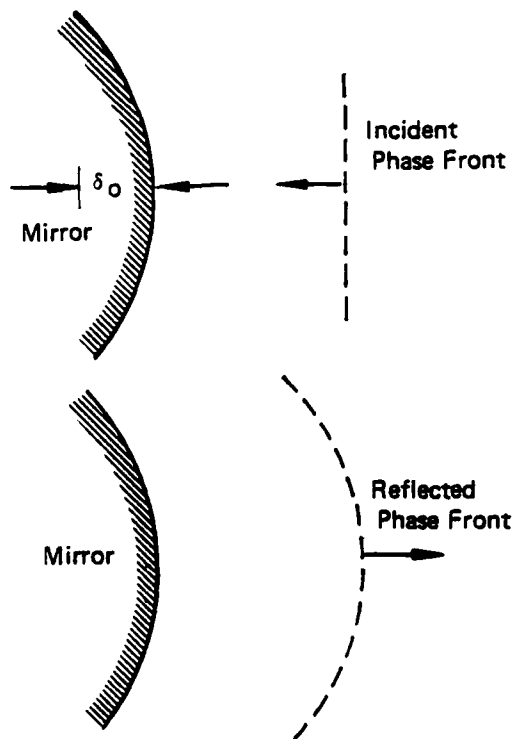


Figure 59. Mirror transmission function relative to the complex field.

Note that the field at the edge has traveled $2\delta_0$ more than the center. The size of the sag $\delta(r)$ (Fig. 40) at any point r can be found from the sag formula:

$$(R_c - \delta)^2 + r^2 = R_c^2 \quad (135)$$

$$\delta \approx \frac{r^2}{2R_c} = \frac{x^2 + y^2}{2R_c}$$

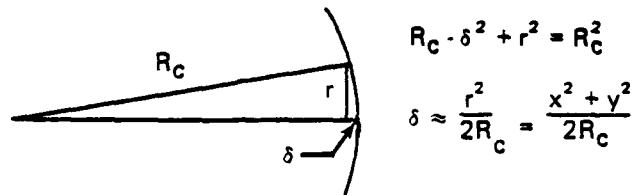


Figure 40. Graphic representation of SAG.

Thus, to make the center of the field lead the edge by a factor of $2\delta_0$, the following transmission function is applied to the field:

$$u'(x,y) = T(x,y) u(x,y), \quad T(x,y) = \varepsilon \frac{2\pi}{\lambda} i \left(\frac{x^2 + y^2}{R_c} \right) \quad (136)$$

The sign convention used is a negative radius of curvature for a convex mirror. A concave mirror has a positive radius of curvature.

In addition to curvature, the MIRROR routine can apply power or flux dependent distortions to the field.

The power dependent mirror distortion can be applied given the center-to-edge maximum sag, DELTA, determined by design power, PWRDES. The incident power is then calculated and the sag reduced by the ratio of incident power to design power. For a ratio greater than one, it is assumed that the sag is that of the design power.

The flux dependence is applied assuming a distortion factor, DISTF, which weights intensity changes from the center of the field and thus applies an intensity-dependent phase factor to the field.

Astigmatism can be applied to the field in conjunction with the localized flux-dependent distortion or can be applied alone. Astigmatism is included if PHIAST is input (as a number greater than 0). PHIAST is the angle between the mirror normal and the optical axis (in degrees). The phase is altered by astigmatism by computing separate (sagittal and tangential) radii of curvature for the mirror and applying to vary the X and Y component of the phase field, respectively.

Argument List

ANX	Mirror tilt in X (about y-axis)
ANY	Mirror tilt in Y (about x-axis)
RADC	Radius of curvature of mirror (cm)
RIAOUT	Outside radius (cm)
RIAIN	Inside radius of annular mirror (cm)
XPOS	X-direction offset of mirror centerline from optical axis of beam (cm)
YPOS	Y-direction offset of mirror centerline from optical axis of beam (cm)
RFL	Mirror reflectivity - fraction 0.0 → 1.0
DELTM	Total power spherical distortion factor
DISTF	Flux distortion factor - local intensity distortion $f(I_{\text{local}} - I_{\text{center}})$; (cm/W/cm ²)
RANULS	Radius to annulus for toric mirror option
RYOUT	Outside Y-dimension (from center) for a rectangular mirror (cm)
RYIN	Inside Y-dimension (from center) for a rectangular mirror (cm)
PHIAST	Angle of beam with respect to mirror normal (deg)

Relevant Variables

AKY	$2\pi/WL = 2\pi/\lambda$ where $\pi = 3.14159$
ANGX	Accumulative x-dim angle to trace field in cavity
ANGY	Accumulative y-dim angle to trace field in cavity

COSP Cosine of phase change
 CUR Real array representing the complete wave amplitude field, i.e., Intensity (J) = $\left[\text{CUR}(J) \right]^2 + \left[\text{CUR}(J-1) \right]^2$
 DELTA DELTM, total power spherical distortion factor
 FMF Square root of mirror reflectivity
 PHASE Phase change at each point of wavefront
 PHI Phase change in TORIC MIRROR and DIMPLE calculations
 PPW Integrated power on mirror
 RADCUR Negative focal length of mirror (-f)
 RMSAG Sagittal radius of curvature (astigmatism)
 RMTAN Tangential radius of curvature (astigmatism)
 SINP Sine of the phase change
 WL Wavelength, λ
 WNDW Magnification factor for scaling power
 XX X^2 ; x-component of location, squared
 YY Y^2 ; y-component of location, squared

Commons Modified

/MELT/

Array modified CUR(I) @ MIRROR 50, 51, 98, 99, 139, 140, 167

/MRPROP/

Variables modified: RADCUR @ MIRROR 115

ANGX @ MIRROR 25

ANGY @ MIRROR 26

EDI (Edge Diffraction Imaging)

Power near the outer edge of the beam that would have been ordinarily lost through diffraction is partially recovered by incorporating a separate radius of curvature in an outer edge annulus, as shown in Equation 137 and Figure 41.

$$\Delta\phi = W_n \left[(x^2 + y^2) - R_{in}^2 \right] / R_{EDI} \quad (137)$$

$$W_n = \frac{2\pi}{\lambda}$$

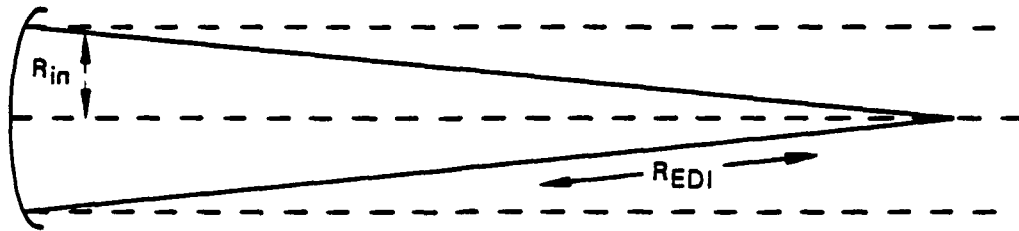


Figure 41. Edge diffraction imaging.

The real representation CUR of the complex amplitude field is modified as follows:

K2 = EVEN NOS

K2MI = ODD NOS

Real Part: CUR (K2) = CUR (K2-1) (sin θ) + CUR (K2) (cos θ)

Im part: CUR (K2MI) = CUR (K2-1) (cos θ) - CUR (K2) (sin θ)

This array is modified in the same way by the phase changes throughout this subroutine.

Mirror Tilt (<100 μ rad)

$$\Delta\phi = -2 \left[(ANX)(X) + (ANY)(Y) \right] \frac{2\pi}{\lambda} \quad (138)$$

where

ANX => tilt in x-direction \sim radians

ANY => tilt in y-direction \sim radians

Power Dependent Spherical Distortion

This part of MIRROR subroutine calculates the phase change due to total power induced spherical distortion.

$$\Delta\phi = \frac{2\pi}{\lambda} \left[\frac{(x^2 + y^2)}{R} \right] \quad (139)$$

where

$$R = R_{out}^2 / 2\delta$$

and $\delta = \text{DELTA} = \text{MAX (Center) DISTORTION} \frac{\text{(Scaled)}}{\text{for Power}}$

$$\text{DELTA} = \text{DELTA}^{(1)} \left(\frac{P_{incident}}{P_{design}} \right)$$

$$R_{out} = \text{Center to edge mirror radius}$$

(1) this is the input DELTA=DELTM

Flux Dependent Distortion (No Astigmatism)

Flux Dependent Distortion + Astigmatism

$$\Delta\phi = \frac{2\pi}{\lambda} \left[\frac{x^2}{R_{SAG}} + \frac{y^2}{R_{TAN}} \right] - \frac{2\pi}{\lambda} \delta_I (1-\text{Ref.})^2 \frac{I_{CL} - I_{xy}}{(\text{WNOW})^2} \quad (140)$$

where

$$R_{SAG} = \text{RADC} / \cos \phi_{ast}$$

and

$$R_{TAN} = \text{RADC} (\cos \phi_{ast})$$

where

RADC = radius of curvature of mirror (cm)

ϕ_{AST} = beam-mirror angle radians = $\text{PHIAST} \frac{\pi}{180}$

I_{CL} = Mirror centerline intensity

I_{XY} = Local (X,Y) intensity

WNOW = VAMP power correction factor

δ_I = Flux distortion factor (cm/W/cm²)

Ref. = Mirror reflectivity

$$\Delta\phi = \frac{-2\pi}{\lambda} \delta_I (1-\text{Ref})^2 \frac{I_{CL} - I_{xy}}{(\text{WNOW})^2} \quad (141)$$

where,

δ_I = Flux distortion factor (cm/W/cm²)

I_{CL} = Intensity at mirror center (W/cm²)

I_{xy} = Intensity at coordinated x,y (W/cm^2)
 Ref = Mirror reflectivity
 WNOW = VAMP power correction factor

Astigmatism (Only)

$$\Delta\phi = \frac{2\pi}{\lambda} \left[\frac{x^2}{R_{SAG}} + \frac{y^2}{R_{TAN}} \right] \quad (142)$$

where $R_{SAG} = RADC / \cos \phi_{ast}$ (cm)
 and $R_{TAN} = RADC (\cos \phi_{ast})$ (cm)
 where RADC = radius of curvature of mirror (cm)
 ϕ_{ast} = beam - mirror (astigmatic) angle
 = PHI_{AST} $\left(\frac{\pi}{180} \right)$

SUBROUTINE MIRROR 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE MIRROR(ANA,ANY,RAUC,HIAOUT,HIAIN,APUS,YPOS,RFL,DELTM, MIRROR 2
X DISTF,HANUL,MYOUT,MYIN,PHIAST) CUASTG 8
C MODIFIED BY JCC 11/4/75 TO MAKE COMPLEX MULTIPLY MORE EFFICIENT. MIRROR 4
C MIRROR TRANSMISSION FUNCTION MIRROR 5
C THIS ROUTINE APPLIES A MIRROR TRANSMISSION FUNCTION TO THE MIRROR 6
C COMPLEX FIELD. THE FOLLOWING EFFECTS ARE INCLUDED: MIRROR 7
C 1. EDGE AND CENTRAL OBSCURATIONS MIRROR 8
C 2. MIRROR RADIUS OF CURVATURE MIRROR 9
C 3. MIRROR REFLECTIVITY MIRROR 10
C 4. POWER DEPENDENT DISTORTION MIRROR 11
C 5. FLUX DEPENDENT DISTORTION MIRROR 12
C 6. TONIC MIRROR RADIUS OF CURVATURE MIRROR 13
C LEVEL 2, CUN MIRROR 14
C COMMON/MLT/CUN(32768),CFIL(16512),A(128),WL,NPTS,NMY,ORA,OMY MIRROR 15
C COMMON/MNPHOP/RAUCUR,ANGA,ANGY MIRROR 16
C COMMON/JAY/WNO,NNEG,NAPTH MIRROR 17
C COMPLEX CFIL CWR2 8
C IF (RIAUI.EQ.0.0.AND.HIAIN.EQ.0.0) GO TO 70 MIRROR 19
C IF (DISTF.LE.-10.) GO TO 300 EUI 1
C CALL APTR(HIAOUT,HIAIN,APUS,YPOS,MYOUT,MYIN) SUAPH 38
C ***** MIRROR 21
C *** MIRROR TILT ADDITION THROUGH STATEMENT NO 50 ***** MIRROR 22
C ***** MIRROR 23
C 70 IF (ABS(ANA) .LE. .000100 .AND. ABS(ANY) .LE. .000100) GO TO 71 MIRROR 24
C ANGX=ANA*2.0 * ANGX MIRROR 25
C ANGY=ANY*2.0 * ANGY MIRROR 26
C 71 DELTA=DELTM MIRROR 27
C FMP=SQRT(RFL) MIRROR 28
C ARY = 2.0 * 3.14159 / WL MIRROR 29
C PI = 3.14159 MIRROR 30
C PWNFAC = 0. MIRROR 31

```

DISMAX = 100000.	MIRROU	32
PPW = 0.	MIRROU	33
PMLT1 = -100000.	MIRROU	34
PWRDES = 500000.	MIRROU	35
IF (MANULS .GT. 0.0) GO TO 100	MIRROU	36
IF (MIAOUT .LT. 0.) GO TO 200	MIRROU	37
IF (ABS(ANX) .GE. .000101 .OR. ABS(ANY) .GE. .000101) GO TO 50	MIRROU	38
IF (ANX .EQ. 0. .AND. ANY .EQ. 0.0) GO TO 50	MIRROU	39
DO 60 J = 1, NPY	MIRROU	40
J1 = (J-1) * NPTS	MIRROU	41
DO 60 I = 1, NPTS	MIRROU	42
TILT = ANX * X(I) + ANY * X(J)	MIRROU	43
PHASE = -2.0 * TILT * ARY	MIRROU	44
K2 = 2 * (I * J1)	MIRROU	45
K2M1 = K2 - 1	MIRROU	46
SINP = SIN (PHASE)	MIRROU	47
CUSP = COS (PHASE)	MIRROU	48
CUMS = CUM(K2M1)	MIRROU	49
CUM(K2M1) = CUMS * COSP + CUM(K2) * SINP	MIRROU	50
CUM(K2) = CUMS * SINP + CUM(K2) * CUSP	MIRROU	51
50 NUB = NPTS * NPY	MIRROU	52
DELMAX = 0.	MIRROU	53
DELIN = 0.	MIRROU	54
C *****	MIRROU	55
C ***** POWER DEPENDENT RADIUS OF CURVATURE CALCULATIONS ARE *****	MIRROU	56
C ***** UN PHASE = 2 PI/LAMUA(A**2 + Y**2/R) *****	MIRROU	57
C ***** R = F(DESIGN POWER, INCIDENT POWER, CENTER TO EDGE DISTORTION) *****	MIRROU	58
C ***** WHERE DESIGN POWER = PWRDES *****	MIRROU	59
C ***** INCIDENT POWER = PPW *****	MIRROU	60
C ***** MAX C. TO E. DISTORTION = DELTA *****	MIRROU	61
C ***** J FORUMAM 11 - 15 - 74 *****	MIRROU	62
C *****	MIRROU	63
DELIN = DELTA	MIRROU	64
IF (DELTA .EQ. 0. .OR. FWF .EQ. 1.) GO TO 30	MIRROU	65
IF (DELTA .LT. 0.) GO TO 20	MIRROU	66
DELMAX = ABS(DELTA)	MIRROU	67
C *****	MIRROU	68
C ***** POWER SCHEDULED CENTER TO EDGE DISTORTION *****	MIRROU	69
C *****	MIRROU	70
DO 15 I = 1, NUB	MIRROU	71
I2 = 2 * I	MIRROU	72
15 PPW = PPW + CUM(I2-1)**2 + CUM(I2)**2	MIRROU	73
PPW = PPW * (X(I2) - X(I1))**2 * (NPTS/NPY)	MIRROU	74
IF (NMEG .EQ. 1 .OR. NMEG .EQ. 2) PPW = PPW / NNUW**2	MIRROU	75
C PWFAC = POWER IN BEAM / DESIGN POWER	MIRROU	76
PWFAC = PPW / PWRDES	MIRROU	77
IF (PWFAC .GT. 1.) PWFAC = 1.	MIRROU	78
DELTA = PWFAC * DELTA	MIRROU	79
GO TO 21	MIRROU	80
20 DELTA = ABS(DELTA)	MIRROU	81
21 RADIUS = (MIAOUT**2) * PI / (WL * ARY * DELTA)	MIRROU	82
RUC = -RADIUS	MIRROU	83
ETA = ARY / RUC	MIRROU	84
RSU = RIAOUT**2	MIRROU	85
DO 23 I = 1, NPY	MIRROU	86
YSU = X(I)**2	MIRROU	87
DO 23 J = 1, NPTS	MIRROU	88
AMG = YSU + X(J)**2	MIRROU	89
IF (AMG .GT. RSU) GO TO 23	MIRROU	90
PHIMIN = ETA * (YSU + X(J)**2)	MIRROU	91
IJ = J * (I - 1) * NPTS	MIRROU	92
IJ2 = IJ * 2	MIRROU	93
IJ2M1 = IJ2 - 1	MIRROU	94
SINP = SIN(PHIMIN)	MIRROU	95
CUSP = COS(PHIMIN)	MIRROU	96
CUMS = CUM(IJ2M1)	MIRROU	97
CUM(IJ2M1) = CUMS * COSP + CUM(IJ2) * SINP	MIRROU	98
CUM(IJ2) = CUMS * SINP + CUM(IJ2) * CUSP	MIRROU	99
23 CONTINUE	MIRROU	100
IF (DELIN .LT. 0.) GO TO 25	MIRROU	101
WRITE(6,89)	MIRROU	102
89 FORMAT(///,2X,35MPWEN SCHEDULED MINMOM DISTORTION	MIRROU	103
WRITE(6,90)RUC, DELMAX, DELTA, PWFAC	MIRROU	104

```

90 FOMAT(//.J9M POWER INDUCED RADIUS OF CURVATURE = .G12.5.2MCM.//, MIRROR 105
X39M MAXIMUM CENTER TO EDGE DISTORTION = .G12.5.2MCM.//, MIRROR 106
X39M APPLIED CENTER TO EDGE DISTORTION = .G12.5.2MCM.//, MIRROR 107
X39M FRACTION OF DESIGN POWER INCIDENT = .G12.5.//) MIRROR 108
GO TO 30 MIRROR 109
25 =NITE (6.91)DELTA.MUC MIRROR 110
91 FOMAT(//.20A.18MMIRROR DISTORTION //, MIRROR 111
X37M APPLIED CENTER TO EDGE DISTORTION = .G12.5.2MCM.//, MIRROR 112
X42M DISTORTION INDUCED RADIUS OF CURVATURE = .G12.5.2MCM) MIRROR 113
30 IF (ABS(MAOC).GT.0.)ARYH=ARY/MAUC MIRROR 114
MAUCM=-MAUC/2. MIRROR 115
IF (PHI.AST .EQ. 0.0) GO TO 350
PHIR = ( PHI.AST * PI)/180.
HMSAG = MAUC / COS(PHIR)
HMTAN = MAUC * COS(PHIR)
350 CONTINUE
NP=NPIS/2
NDEX=(NP-1)*NPIS*NP
IF (FMF.EQ.1.0.AND.UISF.EQ.0.0)GO TO 10
ALPHA = 1.0-FMF**2
DELF=UISF*ALPHA
DELF2=DELF**2.
XICL=CUR(2*NDEX-1)**2 * CUM(2*NDEX)**2
IF (ABS(MAOCUM).LT.5) GO TO 2
WNUWSU=1.0
IF (INKEG .EQ. 1 .ON. NNEG .EQ. 2 ) WNUWSU=WNUW**2
JJ = 0
DO 1 I=1,NPY
YSU = X(I)**2
DO 1 J=1,NPIS
JJ = JJ + 1
JJ2 = JJ * 2
JJ2M1 = JJ2 - 1
XIAY = CUM(JJ2M1)**2 * CUM(JJ2)**2
UELL=DELF2*(XICL-XIAY)/WNUWSU
IF (PHI.AST .EQ. 0.0) GO TO 400
PHASE = ARY*(( X(J)**2 /HMSAG) + (YSU/HMTAN)) -ARY*UELL
GO TO 405
400 PHASE = ARY*(X(J)**2 + YSU ) - ARY*UELL
405 CONTINUE
SINP = SIN(PHASE)
COSP = COS(PHASE)
CUMS = CUM(JJ2M1)
CUM(JJ2M1) = ( CUMS*COSP - CUM(JJ2)*SINP ) * FMF
1 CUM(JJ2) = ( CUMS*SINP + CUM(JJ2)*COSP ) * FMF
IF (PHI.AST.NE.0.0)WITE(6.02)HMSAG,HMTAN
*20 FOMAT(//.---ASTIGMATIC PHASE ABERRATION APPLIED WITH---//,
X20A.---SAGGITAL MIRROR RADIUS= .E15./.*CM.//,
X20A.---TANGENTIAL MIRROR RADIUS= .E15./.*CM.//)
RETURN
2 JJ = 0
DO 3 I=1,NPY
DO 3 J=1,NPIS
JJ = JJ + 1
JJ2 = JJ * 2
JJ2M1 = JJ2 - 1
XIAY = CUM(JJ2M1)**2 * CUM(JJ2)**2
UELL=DELF2*(XICL-XIAY)
PHASE = ARY * ( -UELL )
SINP = SIN(PHASE)
COSP = COS(PHASE)
CUMS = CUM(JJ2M1)
CUM(JJ2M1) = ( CUMS*COSP - CUM(JJ2)*SINP ) * FMF
3 CUM(JJ2) = ( CUMS*SINP + CUM(JJ2)*COSP ) * FMF
RETURN
10 IF (ABS(MAOCUM).LT.5) RETURN
JJ = 0
DO 11 I=1,NPY
YSU = X(I)**2
DO 11 J=1,NPIS
JJ = JJ + 1
IF (PHI.AST.EQ.0.0)GO TO 480

```


IJ2 = (I * K) * 2	MIRROH	202
IJ2M1 = IJ2 - 1	MIRROH	203
SINP = SIN(PH1)	MIRROH	204
COSP = COS(PH1)	MIRROH	205
CUMS = CUM(IJ2M1)	MIRROH	206
CUM(IJ2M1) = CUMS * COSP - CUM(IJ2) * SINP	MIRROH	207
CUM(IJ2) = CUMS * SINP + CUM(IJ2) * COSP	MIRROH	208
205 CONTINUE	MIRROH	209
HMT = -HIAOUT	MIRROH	210
WRITE (6,600) HAU0,HH1	MIRROH	211
600 FORMAT (1/0.20M THE RADIUS OF CURVATURE, E11.0,40M HAS BEEN APPLIED	MIRROH	212
ONLY UPEN A RADIUS OF ,F6.3/)	MIRROH	213
RETURN	MIRROH	214
END	MIRROH	215

18. SUBROUTINE MIX

a. Purpose -- MIX calculates relaxation and pumping rates for use by subroutine KINET. The time constants which describe the various collisional processes are generated from quadratic fits to published data over a finite temperature range. The relaxation rates are then calculated from the time constants and the cavity gas mixture ratio. This routine does not require an argument list.

Relevant Variables

TC2C	time constant for CO ₂ (OVO) + CO ₂ → CO ₂ + CO ₂
TC2N	time constant for CO ₂ (OVO) + CO ₂ → CO ₂ + N ₂
TC2O	time constant for CO ₂ (OVO) + O ₂ → CO ₂ + O ₂
TC2W	time constant for CO ₂ (OVO) + H ₂ O → CO ₂ + H ₂ O
TC3C	time constant for CO ₂ (OVO) + CO ₂ → CO ₂ (OVO) + CO ₂
TC3N	time constant for CO ₂ (OOV) + N ₂ → CO ₂ (OVO) + N ₂
TC3W	time constant for CO ₂ (OOV) + H ₂ O → CO ₂ (OVO) + H ₂ O
TPMP	time constant for N ₂ (V=1) + CO ₂ → N ₂ + CO ₂ (001)
TTRD	$T_s^{-1/3}$
TTRD2	$T_s^{-2/3}$
RC2	relaxation rate for CO ₂ (OVO) → CO ₂ (000)
RC3	relaxation rate for CO ₂ (OOV) → CO ₂ (OVO)
RN2	nitrogen mispump rate (pumps CO ₂ bending mode)
RPUMP	pumping rate for upper level excitation

b. Relevant formalism -- The CO₂ V-V and V-T relaxation rates, the pumping rate and the nitrogen mispump rate are computed by

$$R = P_s \sum_i x_i / \tau_i \quad (143)$$

where P_s is the static pressure,
 X_i are the appropriate species mole fractions,
and τ_i are their associated time constants.

The time constants, τ_i , associated with the various collisional processes are computed by an exponential quadratic fit to published data. The general form is:

$$\tau_i = \exp \left(a_i T_s^{-\frac{2}{3}} + b_i T_s^{-\frac{1}{3}} + c_i \right) \quad (144)$$

Commons Modified

/RATE/

Variables modified:

RC2	at	MIX. 28
RC3	at	MIX. 29
RN2	at	MIX. 30
RPUMP	at	MIX. 31

Commons Modified:

/MELT/

Arrays Modified:

CU incoming & outgoing field. Field is modified.
CFIL field to which CU is made orthogonal
Figure 42 is the subroutine MIX flow chart.

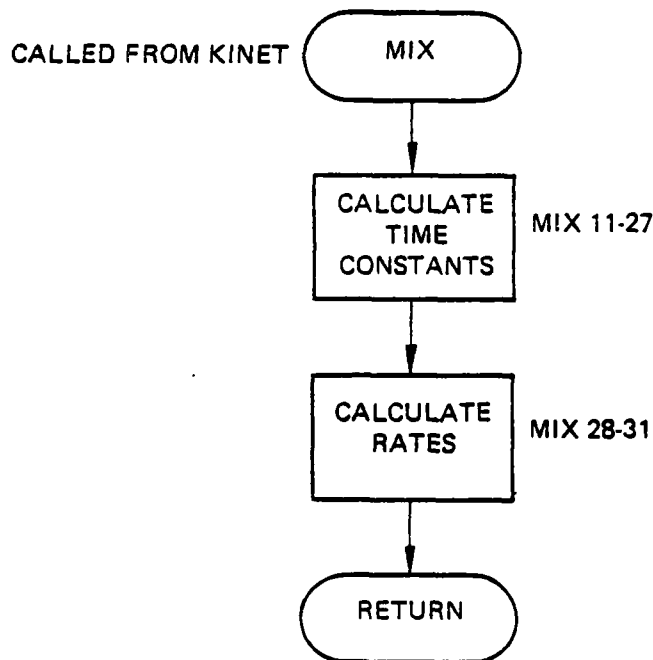


Figure 42. Subroutine MIX flow chart.

The subroutine MIX computer printout follows.

SUBROUTINE MIX 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

	SUBROUTINE MIX	MIX	2
C	THIS ROUTINE CALCULATES THE CO2 V-V AND V-T RELAXATION FOR USE	MIX	3
C	IN SUBROUTINE KINET	MIX	4
	COMMON/PROPT/TS,PS,V,MMU,MMUN,CP,GAMMA,R,B,XLAMB,MNU,CPHM	MIX	5
	COMMON/MULES/XN2,XCO2,XM2U,XCU,XO2	MIX	6
	COMMON/RATE/MN2,MC3,MC2,MPUMP, NSIIM	MIX	7
	TTRU = TS**(-.333)	MIX	8
	TTHO2 = TTRU**2	MIX	9
C	CO2(OOV)*N2 = CO2(OV0)*N2	MIX	10
	TC3N = EXP(-393.12*TTHO2+147.64*TTRU-10./20)	MIX	11
C	CO2(OOV)*O2 = CO2(OV0)*O2	MIX	12
C	TC3O = TC3N	MIX	13
C	CO2(OOV)*CO2 = CO2(OVV)*CO2	MIX	14
	TC3C = EXP(-553.95*TTHO2+200.39*TTRU-15.891)	MIX	15
C	CO2(OOV)*M2U = CO2(OV0)*M2U	MIX	16
	TC3W = EXP(-15.895*TTHO2+.35139*TTRU-2.7323)	MIX	17
C	CO2(OV0)*N2 = CO2*N2	MIX	18
	TC2N = EXP(-294.51*TTHO2+119.88*TTRU-8.6658)	MIX	19
C	CO2(OV0)*CO2 = CO2*CO2	MIX	20
	TC2C = EXP(-295.96*TTHO2+120.32*TTRU-9.3265)	MIX	21
C	CO2(OV0)*M2U = CO2*M2U	MIX	22
	TC2W = EXP(319.24*TTHO2-132.04*TTRU+6.9092)	MIX	23
C	CO2(OV0)*O2 = CO2*O2	MIX	24
	TC2O = EXP(-195.29*TTHO2+86.360*TTRU-6.8646)	MIX	25

C	NZ(V=1)*CO2 = NZ*CU2(001)	MIX	26
	TPMP = EXP(305.25*TTRO2-100.90*TTRO*7.08/1)	MIX	27
	RC2 = PS*(AN2/TC2N*XC02/TC2L*AN20/TC2W*XC02/TC20)*1.E6	MIX	28
	RC3 = PS*(AN2/TC3N*XC02/TC3L*AN20/TC3W*XC02/TC30)*1.E6	MIX	29
	RN2 = PS*XC02/TC3N*1.E6	MIX	30
	RPUMP = PS*(AN2*AC02)/TPMP*1.E6	MIX	31
	RETURN	MIX	32
	END	MIX	33

19. SUBROUTINE MODER:

a. Purpose -- Subroutine MODER is designed to orthogonalize one complex field with respect to another, and to excite a higher order mode for bare resonator mode studies. The fundamental relationships are from the Siegman-Miller paper (Ref. 13). Figures 43, 44, and 45 are flow charts for the Subroutine MODER Organization.

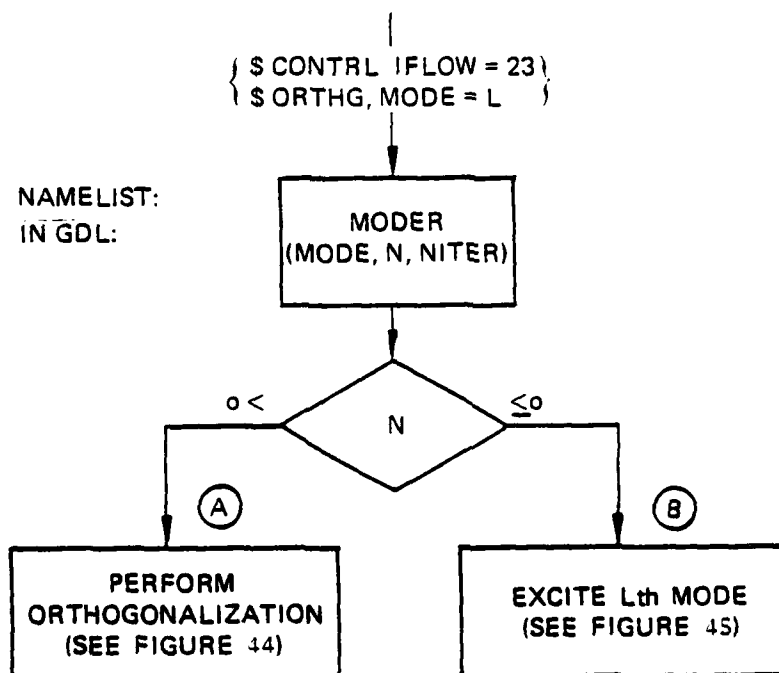


Figure 43. Subroutine MODER organization.

13. Siegman, A. E. and H. Y. Miller, "Unstable Optical Resonator Loss Calculations Using the Prony Method," Applied Optics, 9, p. 2729, 1970.

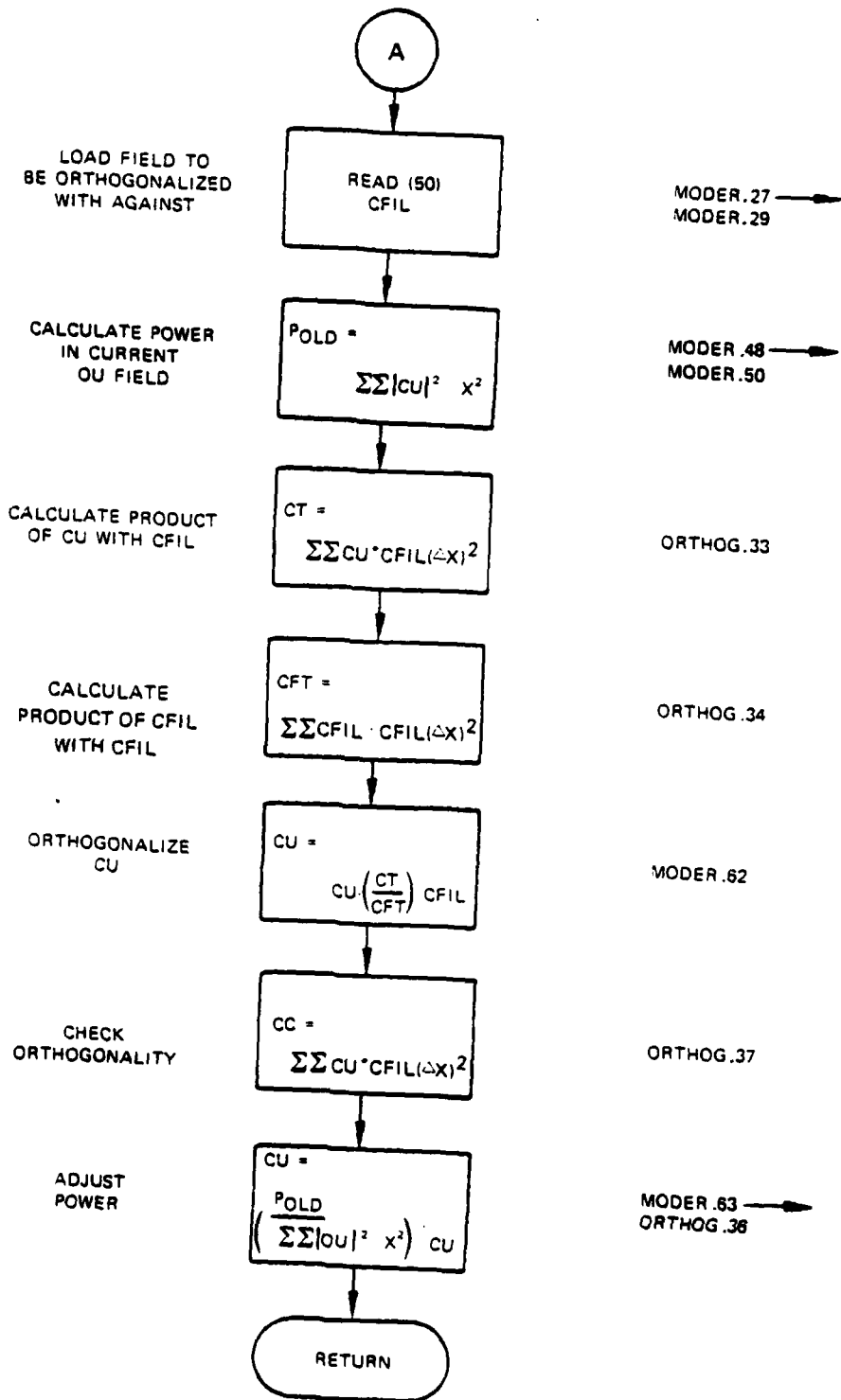


Figure 44. Perform orthogonalization.

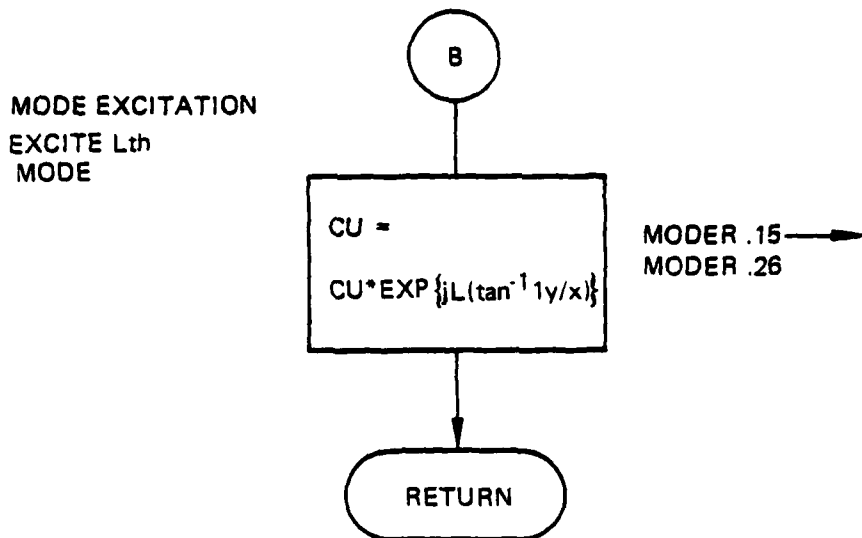


Figure 45. Mode excitation.

b. Relevant formalism -- The orthogonality condition satisfied for symmetric kernel calculations is

$$\iint_R f(x,y)g(x,y) dx dy = 0 \quad (145)$$

where

R = calculation region of interest

f,g = two arbitrary complex fields, described here at equispaced discrete points.

The procedure is implemented by a Gramm-Schmidt orthogonalization, to create a new field, h(x,y) from two known fields. Assume

$$h(x,y) = f(x,y) + cg(x,y) \quad (146)$$

where,

c = complex constant

g = field with which orthogonalization takes place.

then
$$\iint_R dAgh = 0 \quad (147)$$

So
$$c = - \left(\frac{\iint_R fg dA}{\iint_R gg dA} \right)$$

So
$$h = f \left\{ \frac{\iint_R fg dA}{\iint_R gg dA} \right\} g \quad \forall (x,y) \in R$$

Numerically this becomes,

$$h_{ij} = f_{ij} - \left\{ \frac{\sum_i \sum_j f_{ij} g_{ij}}{\sum_i \sum_j g_{ij}^2} \right\} g_{ij} \quad \forall (x_{ij}, y_{ij}) \in R \quad (148)$$

Additionally, impose the condition that

$$h_{ij} = \left[\frac{\iint |f|^2 dA}{\iint |h|^2 dA} \right]^{1/2} h_{ij} \quad (149)$$

then h_{ij} is the new field which is orthogonal with respect to g_{ij} , and has the same power as the initial field f .

Additionally, MODER is structured to excite the azimuthally-varying phase factor for the generation of higher order modes. In cylindrical coordinates, the modes of a bare resonator may be written as:

$$U_{ne}(r, \theta) = \phi_{ne}(r/a) e^{-j1\theta} \quad (150)$$

where,

$$0 \leq \theta \leq 2\pi$$

an arbitrary (convex mirror) scaling factor

$$l = \pm 1, \pm 2, \dots$$

$$n = 0, 1, 2, \dots$$

Higher order modes in bare resonators are initially excited as

$$f'(x,y) = \left[\varepsilon^{-j l \tan^{-1}(y/x)} \right] f(x,y) ; \quad \frac{x^2 + y^2}{a^2} \leq 1 \quad (151)$$

and in discrete form as

$$f_{ij} = \exp \left[-j l \tan^{-1}(y_i/x_i) \right] f_{ij}$$

where f_{ij} is the SOQ complex field distribution.

c. Fortran

Argument List:

N Integer variable denoting the calculation path within the subroutine

N < 0 excite the L^{th} mode and return

N > 0 Perform Orthogonalization

L Order of Mode to be excited

L = 1, 2,

Computer printouts of the MODER subroutine follow.

SUBROUTINE MODER 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE MODER(L,N,N)
C   MODE DISCRIMINATION ROUTINE
C   THIS ROUTINE EXCITES THE L-TH MODE IF N (ITERATION NUMBER) IS 0
C   AND SUPPRESSES LOWER AZIMUTHAL MODES IN SUCCESSIVE ITERATIONS
C
C   ***** THIS COPY DESIGNED TO SUPPRESS L = 0 ONLY   HDQ 11-20-75 ****
C   ***** THIS COPY DESIGNED TO EXCITE L-1ST MODE   HDQ 11-17-75 ***
C
LEVEL 2, CU
COMMON/MELT/CU(16384),CFIL(16512),X(128),WL,NPTS,NPY,DX,DMY
COMMON/AY/WNO,NMLG,HAPTH
COMPLEX CU,CFIL,CT,CFT,CC
PI=3.141592654
IF(N.GT.0) GO TO 100
LP=L-1
DO 10 I=1,NPIS
  XX=X(I)
DO 10 J=1,NPY
  IAJ=(J-1)*NPTS+1
  YY=X(J)
  T=SPATAN(XX,YY) * LP
MODER 2
MODER 3
MODER 4
MODER 5
MODER 6
MODER 7
MODER 8
MODER 9
MODER 10
MODER 11
MODER 12
MODER 13
MODER 14
MODER 15
MODER 16
MODER 17
MODER 18
MODER 19
MODER 20
MODER 21
MODER 22
```

10	CU(I,XJ)=CU(I,XJ)*CEXP(CMPLX(0.0,T))	MODEM	23
	WRITE(6,600) LP	MODEM	24
600	FORMAT(/,10H *** L = ,11.20M MODE HAS BEEN EXCITED ***,/)	MODEM	25
	RETURN	MODEM	26
100	CONTINUE	MODEM	27
	READ(50) (CFIL(I),I=1,NPTS)	MODEM	28
	REWIND 50	MODEM	29
	DO 80 I=1,NPTS	MODEM	30
	DO 80 J=1,64	MODEM	31
	IXJ=(J-1)*NPTS+I	MODEM	32
	IXJ2=(128-J)*NPTS+I	MODEM	33
80	CFIL(IXJ2)=CFIL(IXJ)	MODEM	34
	N0B=NPTS*NPY	MODEM	35
	P=0.0	MODEM	36
	P2=0.0	MODEM	37
	CT=CMPLX(0.0,0.0)	MODEM	38
	CFT=CMPLX(0.0,0.0)	MODEM	39
	DX=ABS(X(1)-X(2))	MODEM	40
	DDU=(DX*W*W)*2*(NPTS/NPY)	MODEM	41
C	WRITE(6,666)DDU,N,CC,CT,AA,P,P2,IFLAG,IFLAG2	MODEM	42
	DO 20 I=1,NPTS	MODEM	43
	DO 20 J=1,NPY	MODEM	44
	IXJ=(J-1)*NPTS+I	MODEM	45
	CT=CT+CONJG(CU(IXJ))*CFIL(IXJ)	MODEM	46
	CFT=CFT+CONJG(CFIL(IXJ))*CU(IXJ)	MODEM	47
20	P=P+CU(IXJ)*CONJG(CU(IXJ))	MODEM	48
	P=P+DDU	MODEM	49
	CC=CT*DDU	MODEM	50
	WRITE(6,604) CC	MODEM	51
604	FORMAT(/,14H *** CC =,2615.5,6M ***,/)	MODEM	52
C	WRITE(6,666)DDU,N,CC,CT,AA,P,P2,IFLAG,IFLAG2	MODEM	53
	CT=CFT/CFT	MODEM	54
	CC=CT	MODEM	55
	WRITE(6,604) CC	MODEM	56
	CC=CFT*DDU	MODEM	57
	WRITE(6,604) CC	MODEM	58
	DO 30 I=1,NPTS	MODEM	59
	DO 30 J=1,NPY	MODEM	60
	IXJ=(J-1)*NPTS+I	MODEM	61
	CU(IXJ)=CU(IXJ)-CT*CFIL(IXJ)	MODEM	62
29	P2=P2+CU(IXJ)*CONJG(CU(IXJ))	MODEM	63
30	CONTINUE	MODEM	64
	P2=P2+DDU	MODEM	65
C	WRITE(6,666)DDU,N,CC,CT,AA,P,P2,IFLAG,IFLAG2	MODEM	66
C	WRITE(6,606) P,P2,N	MODEM	67
C 606	FORMAT(/,14H *** P,P2,N =,3015.5,6M ***,/)	MODEM	68
	SPP=SQRT(P/P2)	MODEM	69
	AA=1.0	MODEM	70
C	AA=SQRT(P/(P-CABS(CT))*2*(NPTS*W))	MODEM	71
	WRITE(6,607) AA,SPP	MODEM	72
607	FORMAT(/,14H *** AA,SPP =,2615.5,6M ***,/)	MODEM	73
	DO 40 I=1,N0B	MODEM	74
40	CU(I)=CU(I)*SPP	MODEM	75
C	WRITE(6,666)DDU,N,CC,CT,AA,P,P2,IFLAG,IFLAG2	MODEM	76
C 666	FORMAT(9H DDU,N =,2615.5,/,9H CC,CT=,3615.5,/,23M P,P2,IFLAG,	MODEM	77
C	IFLAG2 = ,3615.5,2110)	MODEM	78
	RETURN	MODEM	79
	END	MODEM	80

20. SUBROUTINE OUTPUT

a. Purpose -- This routine generates three intensity amplitude and phase printer slice plots through the field. They are along the x-axis, the y-axis, and the "diagonal," defined by the diagram in Figure 46. Figure 47 shows the flow chart for this subroutine.

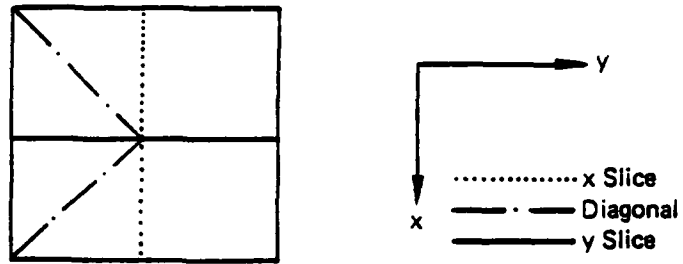


Figure 46. Intensity amplitude and phase printer slice plots.

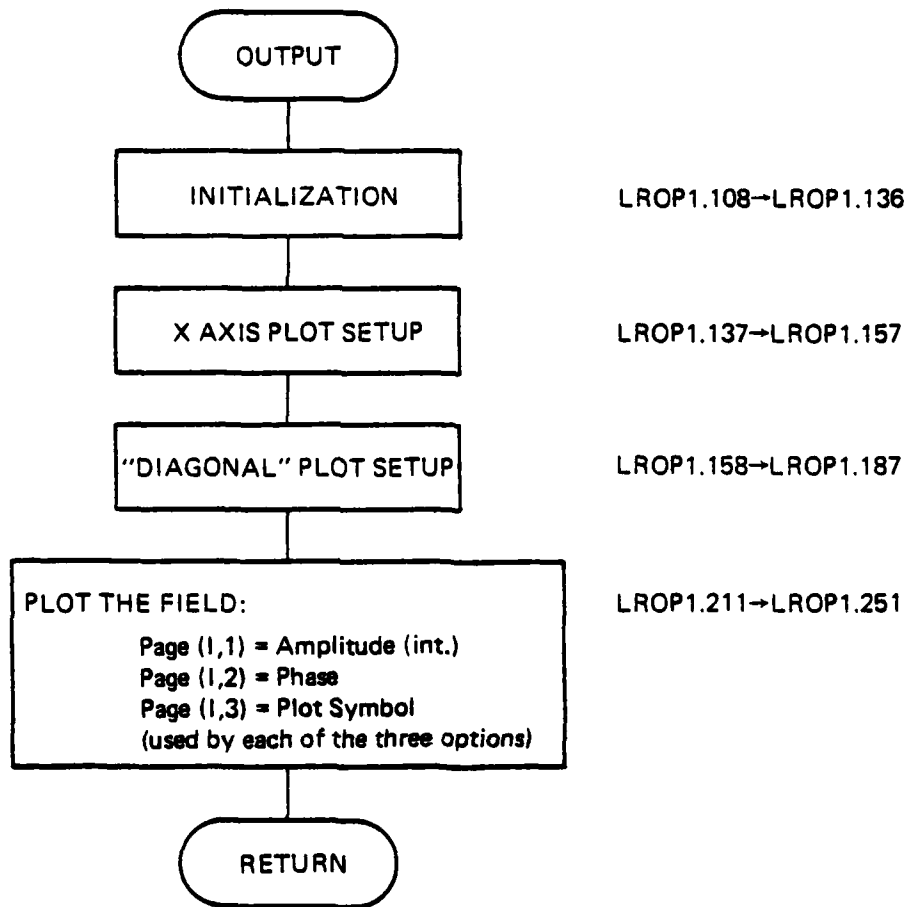


Figure 47. Subroutine OUTPUT flow chart.

b. Relevant formalism -- The slice plot uses 100 available spaces per line for plot information. The point printed shows the percent of maximum amplitude or intensity e.g., if the intensity or amplitude is 35 percent of the maximum, a symbol is printed in the 35th column. Similarly the phase is plotted from -180 to 180 degrees with zero-phase at the center. The corresponding maximum intensity amplitude is also printed out with the appropriate spatial coordinates.

c. Fortran

Argument List

CU field to be plotted
NP2 number of points in the y-direction
NP1 number of points in the x-direction
x coordinate array
N number of plots (1 to 3)
 (N) = 1 → x only
 2 → x and diagonal
 3 → x, diagonal, and y

if $N < 0$, the constant J orders used is $NP/2$ instead of $NP1/2$. This parameter is used when gain/phase slice plots are made.

UMAX - maximum intensity amplitude of the field. It is used to establish the field point to be plotted at 100 percent.

X-AXIS - if true, the x axis plot is generated

DIAG - if true, the "diagonal" plot is generated

Y-AXIS - if true the y axis plot is generated.

No common variables are modified.

No other subroutines are called from this one.

Computer printouts for the OUTPUT subroutine follow.


```

SUBROUTINE OUTPOR(CU,NP2,NP1,X,UMAX,DEG1,DEG2,DEG3)      OUTPUT 2
C NP1=NPTS, NP2=NPY      OUTPUT 3
C THIS ROUTINE CONSTRUCTS PRINTER PLOTS OF RADIAL PROFILES      OUTPUT 4
C AT THREE EQUALLY SPACED ANGLES AROUND THE BEAM      OUTPUT 5
C      OUTPUT 6
LEVEL 2, CU,NP2,NP1,X      OUTPUT 7
COMMON /RAY/ WNUM,NNEG,MAPIN      OUTPUT 8
DIMENSION PAGE(190,J),CU(1),X(1)      OUTPUT 9
COMPLEX CU      OUTPUT 10
LOGICAL DEG1,DEG2,DEG3      OUTPUT 11
C PUT IN PLOTTING SYMBOLS      OUTPUT 12
DATA POINT/1H/,OUT/1H/,BLANK/1H /,APUNT/1H/      OUTPUT 13
XNP2 = NP2      OUTPUT 14
TOTAL = 360. * XNP2 / FLUAT(NP1)      OUTPUT 15
INUX2 = NP2/3      OUTPUT 16
INUX3 = (2 * NP2) / 3      OUTPUT 17
THET1 = TOTAL / 2. / XNP2      OUTPUT 18
THET2 = THET1 * FLUAT(INUX2) * TOTAL / XNP2      OUTPUT 19
THET3 = THET1 * FLUAT(INUX3) * TOTAL / XNP2      OUTPUT 20
100 NP= NP1/2      OUTPUT 21
DO 1000 K=1,3      OUTPUT 22
GO TO (10,20,30),K      OUTPUT 23
10 IF (.NOT.DEG1) GO TO 1000      OUTPUT 24
INDEX = 0      OUTPUT 25
THETA = THET1      OUTPUT 26
GO TO 1      OUTPUT 27
20 IF (.NOT.DEG2) GO TO 1000      OUTPUT 28
INDEX = INUX2 * NP1      OUTPUT 29
THETA = THET2      OUTPUT 30
GO TO 1      OUTPUT 31
30 IF (.NOT.DEG3) GO TO 1000      OUTPUT 32
INDEX = INUX3 * NP1      OUTPUT 33
THETA = THET3      OUTPUT 34
1 DO 410 I = 1,NP1      OUTPUT 35
IREF = I-INDEX      OUTPUT 36
PAGE(I,1) = CAHS(CU(IREF))      OUTPUT 37
DUM1 = AIMAG(CU(IREF))      OUTPUT 38
DUM2 = REAL(CU(IREF))      OUTPUT 39
IF (DUM1.EQ.0.0.AND.DUM2.EQ.0.0) GO TO 412      OUTPUT 40
411 PAGE(I,2) = 57.3*ATAN2(DUM1,DUM2)      OUTPUT 41
GO TO 410      OUTPUT 42
412 PAGE(I,2) = 0.0      OUTPUT 43
410 UMAX = AMAX1(UMAX,PAGE(I,1))      OUTPUT 44
WRITE (6,520) THETA      OUTPUT 45
520 FORMAT(3JH1 CU(I,J) PLOTTED RADially AT ,F7.2,9M DEGREES )      OUTPUT 46
IF (K.NE.1) GO TO 1001      OUTPUT 47
UMAXP=UMAX      OUTPUT 48
IF (NNEG.NE.0.AND.NGT.0)UMAXP=UMAX/WNUM      OUTPUT 49
1001 IF (UMAX.EQ.0.0) UMAX = 1.0      OUTPUT 50
SCALE1 = 100.0/UMAX      OUTPUT 51
SCALE2 = 50.0/180.0      OUTPUT 52
C PRINT AXES      OUTPUT 53
WRITE (6,460)UMAXP      OUTPUT 54
460 FORMAT (1H ,T2,1HU,T27.2M25,152.2M50,158.13MMAGNITUDE (*),T77.2M75      OUTPUT 55
*,T101.5M100 =0G12.4)      OUTPUT 56
WRITE (6,450)      OUTPUT 57
450 FORMAT (1H ,T2,4M=140,T26,3M=90,T52,1MU,158,15MMPHASE ANGLE (*),T76      OUTPUT 58
*,3M=90,T101,4M=180,7A,1MM,7A,4MAMPL,4A,5MMPHASE)      OUTPUT 59
C USE PAGE(L,J) AS PRINTING LINE == FIRST BLANK IT      OUTPUT 60
DO 420 L = 1,104      OUTPUT 61
420 PAGE(L,J) = BLANK      OUTPUT 62
C PRINT A LINE FOR EACH VALUE OF I      OUTPUT 63
DO 430 I = 1,NP1      OUTPUT 64
DO 440 L = 1,101,25      OUTPUT 65
440 PAGE(L,J) = OUT      OUTPUT 66
PAGE(51,3)=SCALE2*PAGE(I,2),J) = APUNT      OUTPUT 67
RELAMP = SCALE1 * PAGE(I,1)      OUTPUT 68
PAGE(1,3) = RELAMP ,J) = PUINI      OUTPUT 69
WRITE (6,470) (PAGE(L,J),L=1,104), X(I),RELAMP ,PAGE(I,2)      OUTPUT 70
470 FORMAT (1H ,10A1,3F9.2)      OUTPUT 71

```

```

PAGE(1.5*SCALE1*PAGE(1,1),3) = BLANK          OUTPUT 72
*30 PAGE(51.5*SCALE2*PAGE(1,2),3) = BLANK     OUTPUT 73
1000 CONTINUE                                  OUTPUT 74
RETURN                                          OUTPUT 75
C.....C                                        OUTPUT 76
END                                             OUTPUT 77

```

SUBROUTINE OUTPUT 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE OUTPUT(CU,NP2,NP1,A,N,UMAX,XXAXIS,DIAG,YAXIS)  LNUP1 108
C NP1=NP15, NP2=NP15 LNUP1 109
C THIS ROUTINE CONSTRUCTS PLOTTER SLICE PLOTS OF THE COMPLEX FIELD LNUP1 110
C ALONG (1) THE Y AXIS, (2) ALONG A DIAGONAL AND (3) ALONG THE LNUP1 111
C X-AXIS THROUGH THE FIELD. Y-AXIS PLOTS ONLY FOR CAVITY PARAMETERS LNUP1 112
LEVEL 2, CU,NP2,NP1,A LNUP1 113
COMMON /WAY/ WNUM,NNEG,NAPIN LNUP1 114
COMMON /PLTSG/ PLTSG LNUP1 115
COMPLEX CU LNUP1 116
LOGICAL XXAXIS,DIAG,YAXIS LNUP1 117
DIMENSION PAGE(190,3),DIAG(128),XP(190),YP(190),CU(1),X(1) LNUP1 118
DIMENSION IMAG(3), IINT(3), ITITL(3) LNUP1 119
C PUT IN PLOTTING SYMBOLS LNUP1 120
DATA POUNI/IM*/.DOT/IMI/,BLANK/IM /,APUNTI/IM*/.PO/INA/IM/ LNUP1 121
DATA IMAG/OMMAGN/OMHITU,OMEL(0)/.IINT/OMINTE,OMNSIT,OMY(0)/ LNUP1 122
IF (PLTSG.GT.0.) GO TO 100 LNUP1 123
POINT = POUNA LNUP1 124
DO 110 IP=1,3 LNUP1 125
110 ITITL(IP) = IMAG(IP) LNUP1 126
GO TO 150 LNUP1 127
100 POINT = POUNI LNUP1 128
DO 120 IP=1,3 LNUP1 129
120 ITITL(IP) = IINT(IP) LNUP1 130
150 CONTINUE LNUP1 131
NP = NP1/2 LNUP1 132
IF (N.LT.0) NP=NP2/2 LNUP1 133
NN=ABS(N) LNUP1 134
DO 1000 K=1,NN LNUP1 135
GO TO (1,2,3),K LNUP1 136
1 IF (.NOT.XXAXIS) GO TO 1000 LNUP1 137
NP2X=NP1*(NP-1) LNUP1 138
C X-AXIS PLOT (I.E. Y=0) LNUP1 139
DO 410 I = 1,NP1 LNUP1 140
XP(I)=X(I) LNUP1 141
YP(I)=X(NP) LNUP1 142
IF (N.LT.0) YP(I)=0.0 LNUP1 143
IREF = I+NP2X LNUP1 144
PAGE(1,1) = CABS(CU(IREF)) LNUP1 145
IF (PLTSG.GT.0.) PAGE(1,1)=PAGE(1,1)**2 LNUP1 146
DUM1 = AIMAG(CU(IREF)) LNUP1 147
DUM2 = REAL(CU(IREF)) LNUP1 148
IF (DUM1.EQ.0.0.AND.DUM2.EQ.0.0) GO TO 412 LNUP1 149
411 PAGE(1,2) = 57.3*ATAN2(DUM1,DUM2) LNUP1 150
GO TO 410 LNUP1 151
412 PAGE(1,2) = 0.0 LNUP1 152
410 UMAX = AMAX1(UMAX,PAGE(1,1)) LNUP1 153
UMAXP=UMAX LNUP1 154
IF (NNEG.NE.0.AND.N.GT.0.AND.PLOTSG.L1.0.) UMAXP=UMAX/WNUM LNUP1 155
IF (NNEG.NE.0.AND.N.GT.0.AND.PLOTSG.GT.0.) UMAXP=UMAX/WNUM**2 LNUP1 156
GO TO 1001 LNUP1 157
2 IF (.NOT.DIAG) GO TO 1000 LNUP1 158
DO 10 I=1,NP1 LNUP1 159
I1=NP1-I-1 LNUP1 160
DIAG(I1)=(MIN0(I1,1)-1)*NP1+1 LNUP1 161
10 CONTINUE LNUP1 162
C DIAGONAL PLOT (I.E. X=Y) LNUP1 163
DO 510 I = 1,NP1 LNUP1 164

```

XP(I)=X(I)	LHUP1	165
NYP=NP1-1	LHUP1	166
IYP=M(INO(I),NYP)	LHUP1	167
YP(I)=X(IYP)	LHUP1	168
IREF = IDIAG(I)	LHUP1	169
PAGE(I,1) = CABS(CU(IREF))	LHUP1	170
IF (PLOTSG.GT.0.) PAGE(I,1)=PAGE(I,1)**2	LHUP1	171
DUM1 = AIMAG(CU(IREF))	LHUP1	172
DUM2 = REAL(CU(IREF))	LHUP1	173
IF (DUM1.EQ.0.0.AND.DUM2.EQ.0.0) GO TO 512	LHUP1	174
511 PAGE(I,2) = 57.3*ATAN2(DUM1,DUM2)	LHUP1	175
GO TO 510	LHUP1	176
512 PAGE(I,2) = 0.0	LHUP1	177
510 IF (PLOTSG.LT.0.) UMAX = AMAX1(UMAX,PAGE(I,1))	LHUP1	178
IF (PLOTSG.GT.0.) GO TO 935	LHUP1	179
WRITE(6,520)	LHUP1	180
520 FORMAT(75M)AMPLITUDE,PHASE PLOTTED ALONG A DIAGONAL THROUGH THE CE	LHUP1	181
XENTER OF UCALC)	LHUP1	182
GO TO 1001	LHUP1	183
935 WRITE(6,934)	LHUP1	184
934 FORMAT(75M)INTENSITY,PHASE PLOTTED ALONG A DIAGONAL THROUGH THE CE	LHUP1	185
XENTER OF UCALC)	LHUP1	186
GO TO 1001	LHUP1	187
C Y=AXIS PLOT (I.E. X=0)	LHUP1	188
3 IF (.NOT.YAXIS) GO TO 1000	LHUP1	189
DO 610 I = 1,NP2	LHUP1	190
XP(I)=X(NP)	LHUP1	191
YP(I)=X(I)	LHUP1	192
IREF = NP*(I-1)*NP1	LHUP1	193
PAGE(I,1) = CABS(CU(IREF))	LHUP1	194
IF (PLOTSG.GT.0.) PAGE(I,1)=PAGE(I,1)**2	LHUP1	195
DUM1 = AIMAG(CU(IREF))	LHUP1	196
DUM2 = REAL(CU(IREF))	LHUP1	197
IF (DUM1.EQ.0.0.AND.DUM2.EQ.0.0) GO TO 612	LHUP1	198
611 PAGE(I,2) = 57.3*ATAN2(DUM1,DUM2)	LHUP1	199
GO TO 610	LHUP1	200
612 PAGE(I,2) = 0.0	LHUP1	201
610 IF (PLOTSG.LT.0.) UMAX = AMAX1(UMAX,PAGE(I,1))	LHUP1	202
IF (PLOTSG.GT.0.) GO TO 3204	LHUP1	203
WRITE(6,620)	LHUP1	204
620 FORMAT(70M)AMPLITUDE,PHASE PLOTTED IN Y-DIRECTION THROUGH CENTER O	LHUP1	205
XF UCALC)	LHUP1	206
GO TO 1001	LHUP1	207
3204 WRITE(6,852)	LHUP1	208
852 FORMAT(70M)INTENSITY,PHASE PLOTTED IN Y-DIRECTION THROUGH CENTER O	LHUP1	209
XF UCALC)	LHUP1	210
1001 IF (UMAX.EQ.0.0) UMAX = 1.0	LHUP1	211
SCALE1 = 100.0/UMAX	LHUP1	212
SCALE2 = 50.0/180.0	LHUP1	213
C PRINT AXES	LHUP1	214
WRITE (6,460) ITITL, UMAXP	LHUP1	215
460 FORMAT (1M ,I2,1M0,I2/,2M25,152,2M50,158,JA4	LHUP1	216
*,I101,5M100 =6I2.4)	LHUP1	217
IF (N.GT.0) WRITE (6,450)	LHUP1	218
450 FORMAT (1M ,I2,4M=180,I26,3M=90,I52,1M0,158,15M)PHASE ANGLE (*),I76	LHUP1	219
*,3M=90,I101,4M=180,IX,1M,4X,1M,4X,1M)	LHUP1	220
IF (N.LT.0) WRITE (6,451)	LHUP1	221
451 FORMAT (1M ,I2,4M=180,I26,3M=90,I52,1M0,158,15M)PHASE ANGLE (*),I76	LHUP1	222
*,3M=90,I101,4M=180,6X,4MANEP,6X,1M)	LHUP1	223
C USE PAGE(L,J) AS PRINTING LINE == FIRST BLANK IF	LHUP1	224
DO 420 L = 1,130	LHUP1	225
420 PAGE(L,J) = BLANK	LHUP1	226
C PRINT A LINE FOR EACH VALUE OF I	LHUP1	227
NEH=NP1	LHUP1	228
IF (K.EQ.3) NEH=NP2	LHUP1	229
IF (N.LT.0) GO TO 301	LHUP1	230
DO 430 I = 1,NEH	LHUP1	231
DO 440 L = 1,I01,25	LHUP1	232
440 PAGE(L,J) = 00T	LHUP1	233
PAGE(13,3)*SCALE2*PAGE(I,2,3) = AMUNT	LHUP1	234
PAGE(1,3)*SCALE1*PAGE(I,1,3) = PMUNT	LHUP1	235

H=SQRT(XP(1)**2+YP(1)**2)	LNUP1	236
WRITE (6,470) (PAGE(L,3),L=1,10),H,XP(1),YP(1)	LNUP1	237
470 FORMAT (1H,10A1,3F9.2)	LNUP1	238
PAGE(1.5*SCALE1*PAGE(1,1),J) = BLANK	LNUP1	239
430 PAGE(51.5*SCALE2*PAGE(1,2),J) = BLANK	LNUP1	240
GO TO 1000	LNUP1	241
301 DO 330 I = 1,NEH	LNUP1	242
DO 340 L = 1,101,25	LNUP1	243
340 PAGE(L,J) = DUT	LNUP1	244
PAGE(51.5*SCALE2*PAGE(1,2),J) = APOINT	LNUP1	245
PAGE(1.5*SCALE1*PAGE(1,1),J) = PPOINT	LNUP1	246
WRITE (6,370) (PAGE(L,J),L=1,10),XP(I),YP(I)	LNUP1	247
370 FORMAT (1H,10A1,2F9.2)	LNUP1	248
PAGE(1.5*SCALE1*PAGE(1,1),J) = BLANK	LNUP1	249
330 PAGE(51.5*SCALE2*PAGE(1,2),J) = BLANK	LNUP1	250
1000 CONTINUE	LNUP1	251
RETURN	LNUP1	252
C.....C	LNUP1	253
END	LNUP1	254

21. SUBROUTINE PLTOT

Subroutine PLTOT is called at the end of subroutine QUAL to calculate and generate a printer plot of far field power versus radial distance in $R\lambda/D$ units. The integrated fractional power to several far field radii are calculated by multiple calls to subroutine POWWOW. The power and radius values are stored by PLTOT in array form. The arrays are then tabulated. A simple printer plot is also generated without the necessity of an interpolation scheme or other formal calculations.

Figure 48 is the Subroutine PLTOT flow chart and is followed by the PLTOT computer printouts.

Argument List

DB	near field beam diameter
DX	grid spacing in far field, $R\lambda/D$ units
IMAX	number of field points across grid
IPLT	flag - not used
PT	total near field power
RMAX	not used
TITLE	run identification
WL	wavelength
XCEN	X-position of center of interest
XX	X-position array
YCEN	Y-position of center of interest
Z	far field intensity array
ZMAX	not used

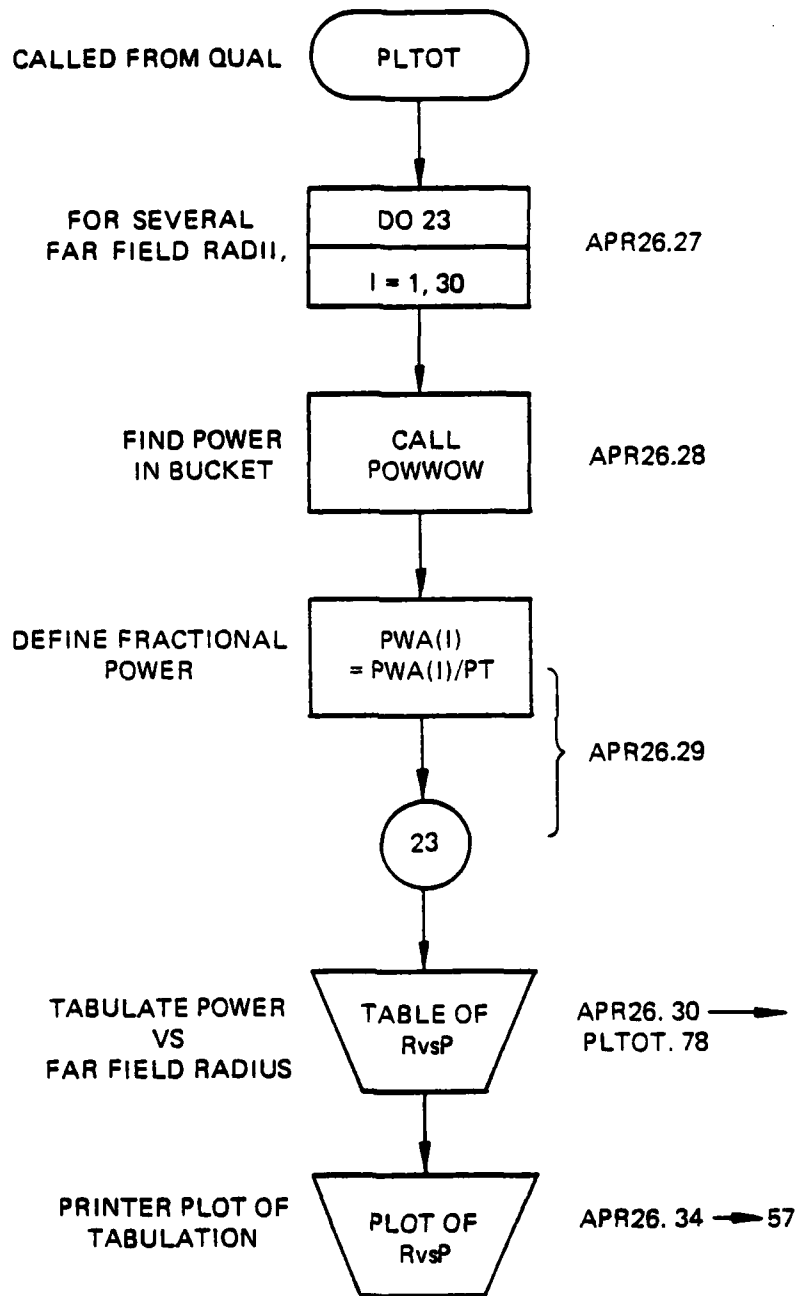


Figure 48. Subroutine PLTOT flow chart.

Relevant variables

IPAGE Hollerith character string comprising a single vertical position of printer plots

PWA fractional power array corresponding to RRD

RRD radial distance array corresponding to various far field bucket sizes.

SUBROUTINE PLTOT 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE PLTOT ( IMAX, UA, XX, ZMAX, HMAX, Z, IPLT, TITLE,      PLTOT      2
1 PT, XCEN, YCEN, UB, NL )                                     APH26      21
LEVEL 2, NL                                                    APH26      22
C THIS ROUTINE (1) MAKES AN ISO-INTENSITY PLOT OF THE FAR FIELD PLTOT      4
C SPOT AND (2) CALCULATES AND PLOTS THE POWER VERSUS FAR FIELD PLTOT      5
C RADIUS.                                                       PLTOT      6
C LEVEL 2, IMAX, XX, Z                                          PLTOT      7
C DIMENSION XX(1), Z( 1 ), TITLE(20)                          PLTOT      8
C DIMENSION PWA(30), HNU(30), IPAGE(101)                       APH26      23
C DATA HNU / 2., 4., 5., 6., 7., 8., 9., 1., 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, APH26      24
X 1., 1.0, 1.9, 2., 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3., 4., 5., / APH26      25
C DATA IBLNK / 4M / .11 / 1M1 / 1PT / 1M. /                  APH26      26
C DIMENSION LAB(5)                                             PLTOT      10
C DATA NUZ, LAB / 6, 80, 60, 40, 20, 10, 5, 2, 1, 12*0 /     PLTOT      11
C CALL DATE(MNTH, DAY, YEAR)                                    PLTOT      12
C CALL MCLK(MH, MIN, SEC)                                       PLTOT      13
C GO TO (32, 51), IPLT                                          PLTOT      14
C PLOT FAR FIELD ISO-INTENSITIES                                PLTOT      15
C XSCL=3./HMAX                                                  PLTOT      16
C CALL INI1(6SIZE, 8., 10.)                                     PLTOT      17
C CALL PLOT(3.5, 3.5, 23)                                       PLTOT      18
C CALL TXSIZ(.2., 13)                                           PLTOT      19
C CALL TXPLT(0., 5., 0., 0)                                       PLTOT      20
C WRITE(98, 1)                                                  PLTOT      21
C 1 FORMAT(30M FAR FIELD ISO-INTENSITY CONTOURS )              PLTOT      22
C CALL TXSIZ(.12., 08)                                           PLTOT      23
C CALL TXPLT(0., 4., 5., 0., 0)                                   PLTOT      24
C WRITE(98, 2) TITLE, HMAX, MNTH, DAY, YEAR, MH, MIN, SEC      PLTOT      25
C 2 FORMAT(1X, 20A4 // 29M THE LARGEST RADIUS PLOTTED =, F4.1, 9MH * LAMB / U / PLTOT      26
C 1 / 5MUATE, A2, 1M /, A2, 1M /, A2, 10A, 5M TIME, A2, 1M, A2, 1M, A2) PLTOT      27
C CALL SYMBOL(0., 0., 15, 3, 0., -1)                             PLTOT      28
C DO 190 I=1, 4                                                  PLTOT      29
C   ADP=.04*I                                                    PLTOT      30
C   AUP=.03*I                                                    PLTOT      31
C   CALL DASH(ADP, AUP)                                           PLTOT      32
C   RH=HMAX*I / 4 * XSCL                                          PLTOT      33
C 190 CALL CIRC(CHADE, MH, ECEN, 0., 0.)                          PLTOT      34
C   CALL NOASH                                                    PLTOT      35
C   CALL ISO( XX, XX, Z, ZMAX, 0., IMAX, IMAX, XCEN, YCEN, XSCL, NUZ, LAB, IMAX) PLTOT      36
C   CALL FINI                                                    PLTOT      37
C   IF (IPLT.EQ.3) GO TO 51                                       PLTOT      38
C   PLOT POWER VS. N*LAMBDA/U. THIS IS DONE ABOUT EITHER THE CENTROID PLTOT      39
C   OR PEAK INTENSITY WHICHEVER DEMONSTRATES MAXIMUM PERFORMANCE. PLTOT      40
C 32 CALL INI1(6SIZE, 8., 10.)                                     PLTOT      41
C   CALL PLOT(1.5, 1., 23)                                       PLTOT      42
C   CALL AXIS(0., 0., 11MHADJUS=NL/U, -11., 0., 0., 0., HMAX/4.) PLTOT      43
C   CALL AXIS(0., 0., 13MPCEN=NT PWER, 13.5, .90, 0., 0., 20.) PLTOT      44
C   CALL GHIU(0., 0., 4., 16, 5., 20)                             PLTOT      45
C   CALL TXSIZ(.15., 09)                                          PLTOT      46
C   CALL TXPLT(2., 8., 0., 0)                                       PLTOT      47
C   WRITE(98, 50) TITLE, MNTH, DAY, YEAR, MH, MIN, SEC          PLTOT      48
C 50 FORMAT(20M FAR FIELD QUALITY (FFT) // 20A4 // 5MUATE, A2, 1M /, A2, 1M PLTOT      49
C 1 /, A2, 10A, 5M TIME, A2, 2(1M, A2))                            PLTOT      50
C   CALL MOVEA(0., 0.)                                           PLTOT      51
C 51 IHAD=HMAX*2.                                                 PLTOT      52
C   PRINT POWER VS. N*LAMBDA/U                                    PLTOT      53
C   WRITE(16, 22) TITLE                                          PLTOT      54

```

C	22	FORMAT(///,1X,20A,///,3X,5(2A,16MH,P(Fraction)),/)	PLTOT	55
C		UU 23 I=1,IMAD	PLTOT	56
C		UU 25 J=1,5	PLTOT	57
C		HNU(J)=1*((I-1)*5.+J)	PLTOT	58
C		CALL POWWOW (IMAX, OX, XX, Z, XCEN, YCEN, HNU(J), PWA(J))	PLTOT	59
C		PWA(J) = PWA(J) / PT	PLTOT	60
C	25	IF (IPLT.LE.1) CALL LINEA(HNU(J)*4./HMAX *PWA(J)*5.)	PLTOT	61
C	23	WRITE(6,24) (HNU(K),PWA(K),K=1,5)	PLTOT	62
		HNUX=0.0	PLTOT	63
		UNHOU=0.1	PLTOT	64
		UU 23 I=1,30	APH26	27
		CALL POWWOW(IMAX,OX,XX,Z,XCEN,YCEN,HNU(I),PWA(I))	APH26	28
	23	PWA(I) = PWA(I) / PT	APH26	29
		UU 25 I=1,6	APH26	30
		J1 = (I-1)*5 + 1	APH26	31
		J2 = J1 + 4	APH26	32
	25	WRITE (6,24) (HNU(K),PWA(K),K=J1,J2)	APH26	33
	24	FORMAT(5(4X,F4.1,F8.5))	PLTOT	78
	26	CONTINUE	PLTOT	79
C		IF (IPLT.LE.1) CALL FINI	PLTOT	80
		WRITE(6,1100) W,OB	APH26	34
	1100	FORMAT(1M1///,60X,18MPERCENT TOTAL FLUX /45X,3HNL=F8.6,4M O=	APH26	35
		X,F8.2/2X,6HHD 0.23X,2H25,23A,2H50,23A,2H75,22X,3H100)	APH26	36
		DU 1310 I=2,100	APH26	37
	1310	IPAGE(I) = IBLNK	APH26	38
		IMAD = 1	APH26	39
		DU 1320 LINE=1,51	APH26	40
		IPAGE(1)=I1	APH26	41
		IPAGE(26)=I1	APH26	42
		IPAGE(51)=I1	APH26	43
		IPAGE(76)=I1	APH26	44
		IPAGE(101)=I1	APH26	45
		RAU = (LINE-1)*.1	APH26	46
		PCH = HNU(IMAD)	APH26	47
		IF (ABS(IMAD-PCH).GT.0.01) GO TO 1315	APH26	48
		INDEX = 1.5 * PWA(IMAD)*100.	APH26	49
		IMAD = IMAD + 1	APH26	50
		IPAGE(INDEX) = IPT	APH26	51
		WRITE(6,1110) HAD,IPAGE	APH26	52
	1110	FORMAT (1X,F4.2,2X,101A1)	APH26	53
		IPAGE(INDEX)=IBLNK	APH26	54
		GO TO 1320	APH26	55
	1315	WRITE (6,1110) HAD,IPAGE	APH26	56
	1320	CONTINUE	APH26	57
		RETURN	PLTOT	83
		END	PLTOT	84

22. SUBROUTINE POWWOW

Calls: N/A

Called by: QUAL

a. Purpose -- POWWOW is called by QUAL to apply an aperture to the far field intensity field for computing power in the bucket. Figure 49 shows the POWWOW flow chart, followed by the POWWOW computer printouts.

POWWOW passes the intensity field, x and y centroid locations, and bucket size. It returns the power in the bucket in parameter PRB (PWR).

POWOW defines a radius function, RD, for converting rectangular coordinates to a radius bucket size. Each x,y coordinate is searched to determine if it is within the bucket. If so, the power at that point is added to the sum for the bucket.

After all locations have been checked, control is returned to QUAL along with the power number.

b. Relevant formalism -- Each grid point (X,Y) lies at the center of a square ΔX on a side. In the logic to determine whether a point falls within the radius of interest, an attempt is made to account for grids which fall partially within the radius, RAD. These points are weighted between 0 and 1 according to

$$P = (\text{RAD} - R_{\min}) / (R_{\max} - R_{\min})$$

where

P is the weighting factor,

R_{\max} is the radius to the furthest corner of the grid, and

R_{\min} is the radius to the nearest corner of the grid.

All grid points with R_{\max} less than RAD are given a weight of 1, all grid points with R_{\min} greater than RAD are weighted 0.

Argument List

AA	far field intensity array
DX	separation of far field points
NPTS	number of array points in one dimension
PWR	power in the bucket - returned to calling routine
RAD	radius of far field bucket
XAR	X-position array for intensity field
XCEN	X-position of center of interest
YCEN	Y-position of center of interest

Relevant Variables

DS	area associated with a grid point
PER	weighting factor for a grid point, between 0 and DS
X	X-position of grid point
Y	Y-position of grid point

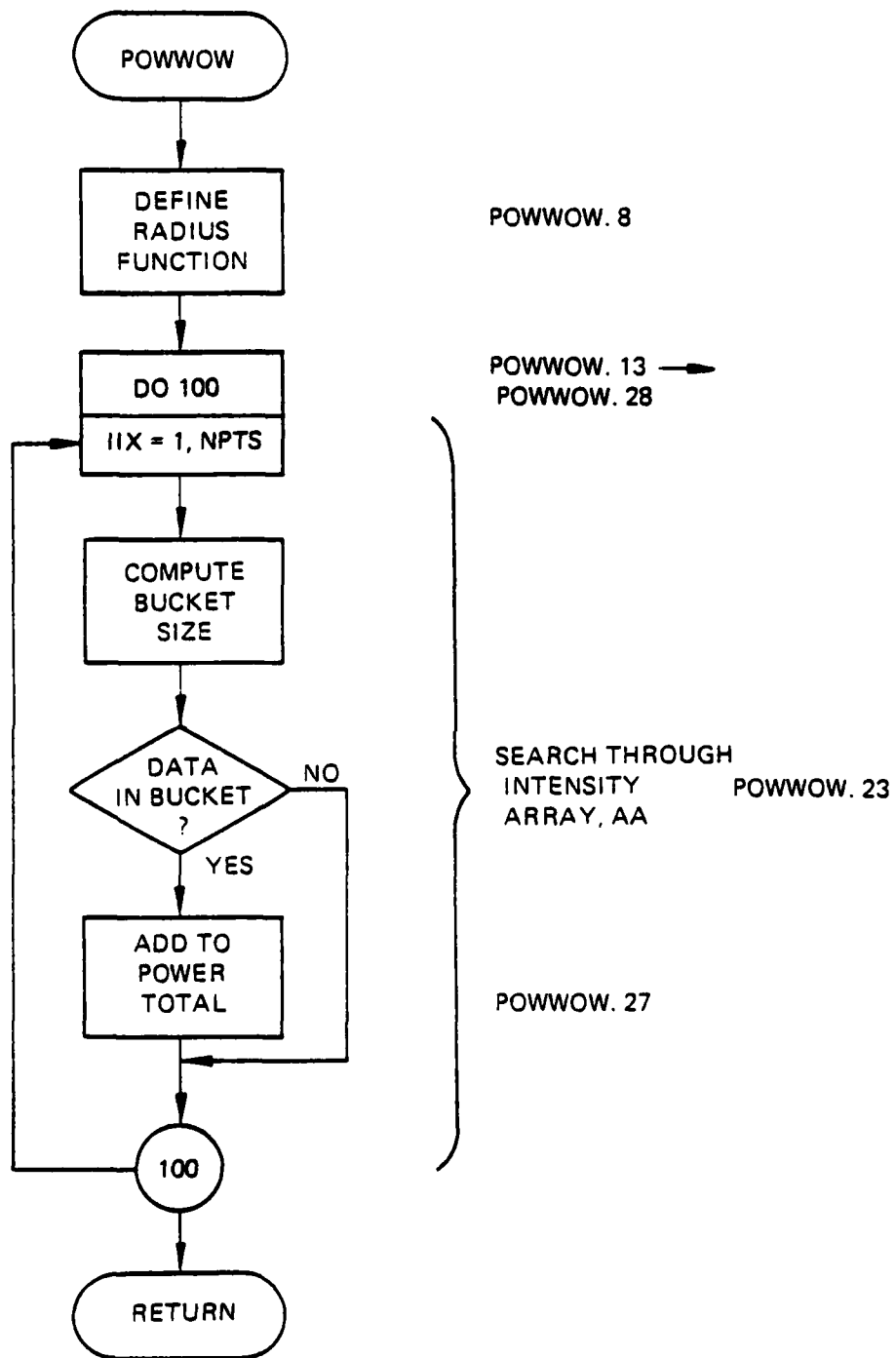


Figure 49. Subroutine POWWOW flow chart.

```

SUBROUTINE POWWOW (  NPTS,  OX,  XAN,  AA,  POWWOW  2
1  XCEN,  YCEN,  RAD,  PWR )  POWWOW  3
C  THIS ROUTINE APPLIES AN APERTURE TO THE FAR FIELD INTENSITY  POWWOW  4
C  PATTERN FOR DETERMINING POWER VS. W*LAMBDA/D  POWWOW  5
LEVEL 2, NPTS, XAN, AA  POWWOW  6
DIMENSION XAN(1), AA( 1 )  POWWOW  7
MU(XA, Y, IX, IY)=SUM((ABS(XA)*[X*OX/2.]**2+(ABS(Y)*OY/2.*[Y]**2)  POWWOW  8
PWR = 0.  POWWOW  9
OY=OX  POWWOW 10
OS = OX ** 2  POWWOW 11
DO 100 IIX=1,NPTS  POWWOW 12
X=XAN(IIX)-XCEN  POWWOW 13
DO 100 IYY=1,NPTS  POWWOW 14
Y=XAN(IYY)-YCEN  POWWOW 15
HPP=HD(X,Y,1,1)  POWWOW 16
HMM=HD(X,Y,-1,-1)  POWWOW 17
HNP=HD(X,Y,-1,1)  POWWOW 18
HMP=HD(X,Y,1,-1)  POWWOW 19
PEH=US  POWWOW 20
HMAX=AMAX1(HPP,HMM,HNP,HMP)  POWWOW 21
IF (HMAX.LE.RAD) GO TO 100  POWWOW 22
PER = 0.  POWWOW 23
HMIN=AMIN1(HPP,HMM,HNP,HMP)  POWWOW 24
IF (RMIN.GE.RAD) GO TO 100  POWWOW 25
PER=(RAD-HMIN)/(HMAX-HMIN)*OS  POWWOW 26
100 PWR=PWR+AA(IIX*(IYY-1)+NPTS)*PER  POWWOW 27
RETURN  POWWOW 28
END  POWWOW 29

```

23. SUBROUTINE QUAL

```

Called by:  MAIN
Calls:     TILT
           STEP
           CENBAR
           POWWOW

```

QUAL, entered with a call from MAIN, is used to calculate quality of complex field. Figure 50 is the flow chart for the QUAL subroutine. Subroutine QUAL computer printouts follow Figure 50. A decision is made whether to use the COMMON complex field or whether to read one in from tape. A decision is then made as to whether or not to save whatever input complex field is used. This is for later restoration.

Variables are initialized and, based on the call statement input variables, a decision is made whether or not to apply a phase correction to the complex field, that is, should tilt and/or spherical components be removed? If not, QUAL branches to the lens section.

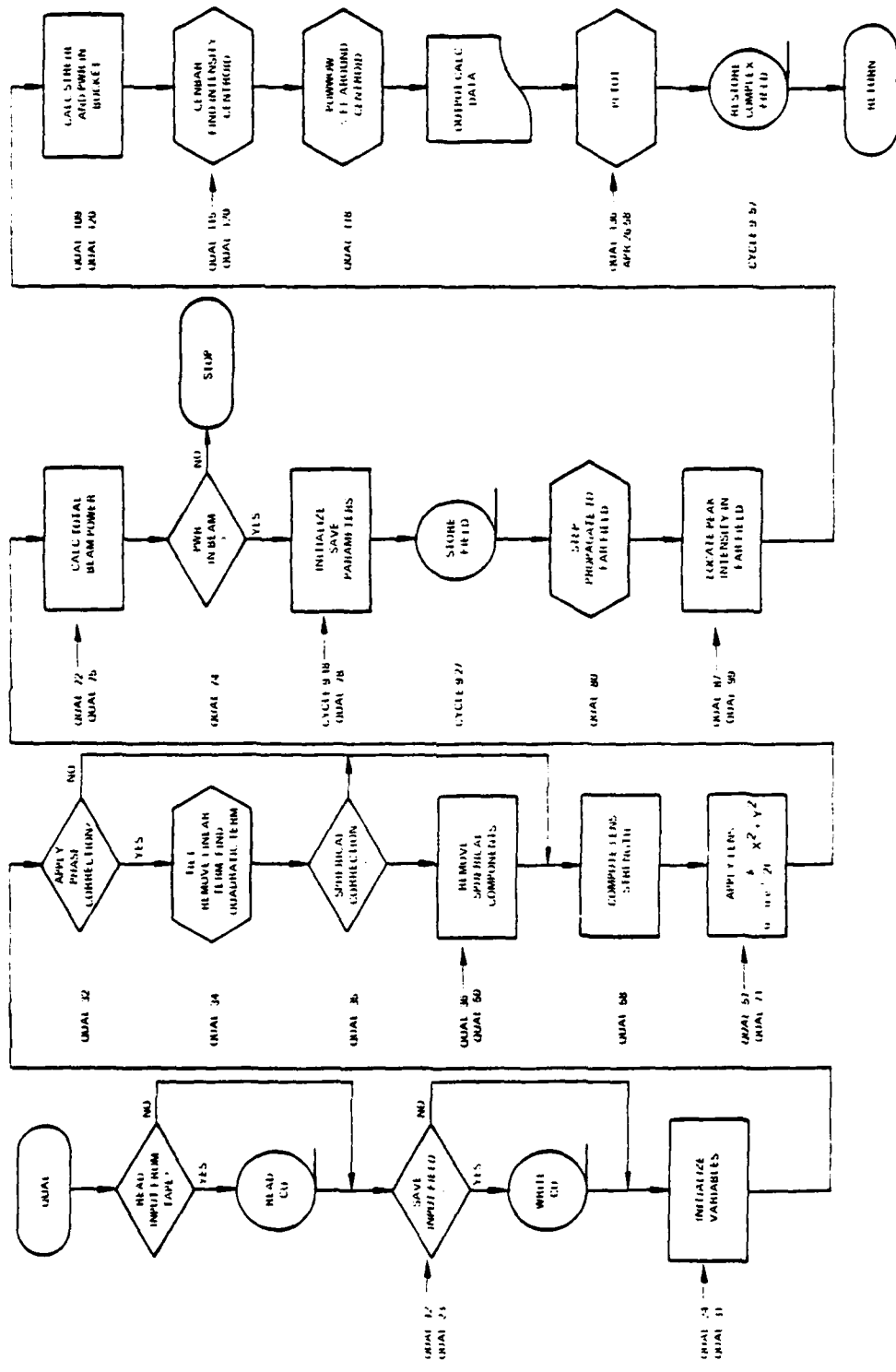


Figure 50. Subroutine QUAL flow chart.

If yes, then a call is made to subroutine TILT and the linear and quadratic phase components are removed. If spherical components are to be removed, then this is done. If not, control passes to the lens section.

The lens strength required to bring the beam down to a specified radius is computed. This is then applied to the field, CU, via the relation

$$U = U \exp \left[i \frac{k}{2f} (x^2 + y^2) \right] \quad (152)$$

The total beam power as transformed by the lens is then calculated. If there is no power in the transformed beam, an error message is output and the job stopped. Otherwise, some saving parameters are initialized and the transformed field is saved on tape.

Subroutine STEP is called to take the transformed beam to the far field. The location of the far field peak intensity is found. Strehl and power in the bucket are calculated. Subroutine CENBAR is called to find the percent of far field centroid (intensity). Subroutine POWWOW is called to find the percent of far field power in a given radius around the centroid. All of the calculated data is printed and subroutine PLTOT is called for beam quality plots.

QUAL then restores the complex field to what it was at entry and control is returned to MAIN.

SUBROUTINE QUAL 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

	SUBROUTINE QUAL (IPHASE , ISAVE , IPLT , TITLE , NB , ANS , UB , NF)	QUAL	2
C	FAR FIELD QUALITY ALGORITHM	QUAL	3
C	THIS ROUTINE IS RESPONSIBLE FOR CALCULATING THE QUALITY OF THE	QUAL	4
C	COMPLEX FIELD.	QUAL	5
	LEVEL 2, CU, CUR, US	QUAL	6
	COMMON/MLT/CU(16384),CFIL(16512),A(128),NL,NPTS,NPY,UMX,UMY	QUAL	7
	DIMENSION TITLE(1),US(16384),ANS(1),CUR(32768)	QUAL	8
	X,FUM(5),P(6),XSAVE(128)	CYCLE9	17
	COMPLEX CU,CFIL,CUNE,CJ,CZEMU	QUAL	9
	EQUIVALENCE (CFIL(1),US(1)) , (CUR(1),CU(1))	QUAL	10
C	***** SAVE FIELD	QUAL	11
	NP=NPTS/2	QUAL	12
	NUM=NPTS*NPY	QUAL	13
	IF (ISAVE.LT.9) GO TO 212	QUAL	14
	HEAD(9) (CU(12),I2=1,NOB),A,UMX,UMY	QUAL	15
	RE=IND 9	QUAL	16
	GO TO 310	QUAL	17

212	IF (ISAVE.NE.1) GO TO 211	QUAL	18
	WRITE(7) (CU(I2),I2=1,NUB),X,UMX,UMY	QUAL	19
	REWIND 7	QUAL	20
211	IF (ISAVE.NE.-1) GO TO 210	QUAL	21
	HEAD(9) (CU(I2),I2=1,NUB),X,UMX,UMY	QUAL	22
	REWIND 9	QUAL	23
210	CONE=(1.UE0,U.UE0)	QUAL	24
	PI=3.141593	QUAL	25
C	CJ=(U.E0+1.E0)	QUAL	26
	CZERO=(0.E0,U.E0)	QUAL	27
	KR=2.*PI/WL	QUAL	28
	AX=0.	QUAL	29
	AY=0.	QUAL	30
	OCAL=X(NPTS)-X(1)+X(2)-X(1)	QUAL	31
	IF (IPHASE.EU.0) GO TO 50	QUAL	32
C	CORRECT LINEAR AND QUADRATIC COMPONENTS OF THE PHASE.	QUAL	33
	CALL TILT(AX,AY,RADIUS,IPHASE)	QUAL	34
	IF (IPHASE.LT.2) GO TO 50	QUAL	35
	BHALF=PI/(WL*RADIUS)	QUAL	36
	DO 65 J=1,NPY	QUAL	37
	J1=(J-1)*NPTS	QUAL	38
	YSU = X(J) **2	QUAL	39
	DO 65 I=1,NPTS	QUAL	40
	IJ=I-J1	QUAL	41
	IJ2 = 2 * IJ	QUAL	42
	IJ2M1 = IJ2 - 1	QUAL	43
	PHI = BHALF * (X(I1)**2 + YSU)	QUAL	44
	SINP = SIN(PHI)	QUAL	45
	COSP = COS(PHI)	QUAL	46
	CUMS = CUM(IJ2M1)	QUAL	47
	CUM(IJ2M1) = CUMS*COSP - CUM(IJ2)*SINP	QUAL	48
	CUM(IJ2) = CUMS*SINP + CUM(IJ2)*COSP	QUAL	49
C 65	CU(IJ)=CU(IJ)*CEXP(CMPLX(U.,BHALF*(X(I1)**2+X(J)**2))	QUAL	50
50	CONTINUE	QUAL	51
C ***	STRENGTH OF LENSE REQUIRED TO KEEP BEAM WITHIN 2.* NR AT FOCUS	QUAL	52
	F = OCAL*DB/(2.*NR*WL)	QUAL	53
	UX=X(2)-X(1)	QUAL	54
	UXS=UX**2	QUAL	55
	PT=0.	QUAL	56
	I2=0	QUAL	57
C	APPLY LENSE TO COMPLEX FIELD	QUAL	58
	DO * M=1,NPY	QUAL	59
	YSU = X(M) **2	QUAL	60
	DO * N=1,NPTS	QUAL	61
	I2=I2+1	QUAL	62
	I22 = 2 * I2	QUAL	63
	I22M1 = I22 - 1	QUAL	64
	PHI = KR * (X(N)**2 + YSU) / 2. / F	QUAL	65
	SINP = SIN(PHI)	QUAL	66
	COSP = COS(PHI)	QUAL	67
	CUMS = CUM(I22M1)	QUAL	68
	CUM(I22M1) = CUMS*COSP - CUM(I22)*SINP	QUAL	69
	CUM(I22) = CUMS*SINP + CUM(I22)*COSP	QUAL	70
C	CU(I2)=CU(I2)*CEXP(CJ*KR*(X(N)**2+X(M)**2)/2./F)	QUAL	71
	* PT = PT + CUM(I22M1)**2 + CUM(I22)**2	QUAL	72
C	* PT=PT+CU(I2)*CUNJG(CU(I2))	QUAL	73
	IF (PT .LE. 0.0) GO TO 200	QUAL	74
	PWSAVE = PT * OXSU * NPTS / NPY	QUAL	75
	DO 295 I=1,NPTS	CYCLE9	18
	XSAVE(I) = X(I)	CYCLE9	19
295	CONTINUE	CYCLE9	20
	OX2SVE=OX*UX	CYCLE9	21
	OXSVE=UX	CYCLE9	22
	PTSVE=PT	CYCLE9	23
	DO 300 I=1,5	CYCLE9	24
	D=(I-3)/10.	CYCLE9	25
300	FOM(I) = F*(1.+D)	CYCLE9	26
	WRITE(1) (CU(IJ),IJ=1,NUB),X,UMX,UMY	CYCLE9	27
	REWIND 1	CYCLE9	28
	ISTEP=0	CYCLE9	29
325	ISTEP=ISTEP + 1	CYCLE9	30
	PT=PTSVE	CYCLE9	31

	DX=OASAVE	CYCLE9	32
	DXSQ =OASVE	CYCLE9	33
	DO 220 I =1,NPTS	CYCLE9	34
	X(I) = XSAVE(I)	CYCLE9	35
220	CONTINUE	CYCLE9	36
	IF(ISTEP.EQ.6) GO TO 335	CYCLE9	37
	F = FBM(ISTEP)	CYCLE9	38
	GO TO 340	CYCLE9	39
335	F=FOPT	CYCLE9	40
340	CONTINUE	CYCLE9	41
	PWSAVK = PWSAVE/1000.	QUAL	76
	ZLD=F*WL/DB	QUAL	77
	P1=PT*DXSQ/(ZLD*ZLD) * NPTS / NPY	QUAL	78
C	PHUPAGATE TO THE FAN FIELD	QUAL	79
	CALL STEP (F,1.0, 0.0,0.1,1.1,0.0,0.0,1.0)	QUAL	80
C	CHANGE X TO FAN FIELD X	QUAL	81
	DO 11 I=1,NPTS	QUAL	82
11	X(I)=X(I)/ZLD	QUAL	83
	DX=DX/ZLD	QUAL	84
	DXSQ=DX*DX	QUAL	85
	UMAX=0.	QUAL	86
C	LOCATE PEAK INTENSITY IN FAN FIELD	QUAL	87
310	DO 61 J=1,NPY	QUAL	88
	J1=(J-1)*NPTS	QUAL	89
	DO 61 I=1,NPTS	QUAL	90
	I2=I+J1	QUAL	91
	I22 = I2 * 2	QUAL	92
	US(I2) = CUM(I22-1)**2 * CUM(I22)**2	QUAL	93
C	US(I2) = CUM(I2) * CUM(CUM(I2))	QUAL	94
	IF (US(I2).LT.UMAX) GO TO 61	QUAL	95
	XPEAK=X(I)	QUAL	96
	YPEAK=Y(J)	QUAL	97
	UMAX=US(I2)	QUAL	98
61	CONTINUE	QUAL	99
	IF(NPTS.EQ.NPY)GO TO 63	QUAL	100
	DO 62 J=1,NPY	QUAL	101
	JJ = NPTS+1-J	QUAL	102
	J1=(J-1)*NPTS	QUAL	103
	DO 62 I=1,NPTS	QUAL	104
	I2=I+J1	QUAL	105
62	US(I+(JJ-1)*NPTS) = US(I2)	QUAL	106
63	UMAX=UMAX/1000.	QUAL	107
	UMX1=PWSAVE*P1*(DB/(WL*F))**2/4.0	QUAL	108
C	STHEML INTENSITY	QUAL	109
	STHEML=UMAX/UMX1	QUAL	110
C	CALCULATE PERCENT OF FAN FIELD POWER WITHIN RB RADIUS OF IPEAK	QUAL	111
	CALL POWW0W(NPTS,DX,X,US,XPEAK,YPEAK,MB,PRH)	QUAL	112
	PRH = PRH * ZLD**2	QUAL	113
	PRK = PRH/1000.	QUAL	114
	P(ISTEP)=PRK	CYCLE9	42
C	LOCATE INTENSITY CENTROID IN FAN FIELD	QUAL	115
	CALL GENBAR (NPTS, DX, X, US, XCINT, YCINT, UMAX)	QUAL	116
C	CALCULATE PERCENT OF FAN FIELD POWER WITHIN RB RADIUS OF CENTROID	QUAL	117
	CALL POWW0W(NPTS,DX,X,US,XCINT,YCINT,MB,PRH)	QUAL	118
	PRH = PRH * ZLD**2	QUAL	119
	PRK = PRH/1000.	QUAL	120
	IF (ISTEP.EQ.6) GO TO 5904	CYCLE9	43
	IF (ISTEP.EQ.1) WRITE(6,5910)	CYCLE9	44
5910	FORMAT (25A,19M)FLUX IN INL/U ABOUT /	CYCLE9	45
	A 20M TRIAL FOCAL LENGTHS, 9X,10MTOTAL DCALC FLUX ,	CYCLE9	46
	X 9X,0MIMAX,9A,0MCENTROID)	CYCLE9	47
	WRITE (6,5920) ISTEP,F,PWSAVK,PRK,PRB	CYCLE9	48
5920	FORMAT (3M F,11,1M,0G12.4,10A,F7.2,12X,F7.2,8X,F7.2)	CYCLE9	49
	GO TO 5930	CYCLE9	50
5904	WRITE(6,5940) F	CYCLE9	51
5940	FORMAT(22M OPTIMUM RESULTS AT F,0G12.4)	CYCLE9	52
	WRITE(6,132) MB,PRK,XCINT,YCINT,MB,PRK,UMXK,XPEAK,YPEAK,PWSAVK,DB	QUAL	121
132	FORMAT(//15M DCALC FLUX IN ,F5.2,6M RL/D=0G12.4,27M ABOUT CENTROID	QUAL	122
	A COORDINATES,2G12.4//15M DCALC FLUX IN ,F5.2,6M RL/D=0G12.4,14M AB	QUAL	123
	XOUT IMAX ,0G12.4,12M CUORDINATES,2G12.4//15M TOTAL DCALC FLUX=,	QUAL	124
	XG12.4,22M REFERENCE DIAMETER=,F6.2)	QUAL	125

WRITE(6,133)STHEML	QUAL	126
133 FORMAT(1/19H STHEML INTENSITY =.G11.4)	QUAL	127
5930 CONTINUE	CYCLE9	53
IF(ISTEP.LE.5) GO TO 305	CYCLE9	54
ANS(1) = PWB	QUAL	128
ANS(2) = PWSAVE	QUAL	129
ANS(3) = UMAX	QUAL	130
IF (PWB.GT.PWB) GO TO 53	QUAL	131
XCINT = XPEAK	QUAL	132
YCINT = YPEAK	QUAL	133
ANS(1) = PWB	QUAL	134
C MAKE SPECIFIED FWH FIELD PLOTS AND CALCULATE POWER VS. R*LAMBDA/D	QUAL	135
53 IF (IPLT.NE.0) CALL PLTOT(NPTS, UA, A, UMAX, 4., US, IPLT,	QUAL	136
A TITLE,PI,XCINT,YCINT,UB,UL)	APM26	58
C ***** RESTORE FIELD	QUAL	138
305 CONTINUE	CYCLE9	55
IF(ISTEP.GE.6) GO TO 350	CYCLE9	56
HEAD(1) (CU(1J),J=1,NOB),A,DX,DRY	CYCLE9	57
RE=INU 1	CYCLE9	58
IF(ISTEP.LT.5) GO TO 325	CYCLE9	59
POPT=-100	CYCLE9	60
DU 375 I=1.5	CYCLE9	61
IF(P(1).LE.PUMT) GO TO 375	CYCLE9	62
PUMT=P(1)	CYCLE9	63
PUMT=FBM(1)	CYCLE9	64
375 CONTINUE	CYCLE9	65
GO TO 325	CYCLE9	66
350 CONTINUE	CYCLE9	67
IF (ISAVE.NE.1) RETURN	QUAL	139
HEAD(1) (CU(1Z),IZ=1,NOB),A,DX,DRY	QUAL	140
RE=INU	QUAL	141
RETURN	QUAL	142
200 WRITE(6,201)	QUAL	143
201 FORMAT(30HNO POWER IN BEAM - JOB KILLED)	QUAL	144
STOP	QUAL	145
END	QUAL	146

24. SUBROUTINE REGAIN

Called from: GDL

Calls: BLUMIT, CPUTIM, FUHS, GAINXY, ISOCAV, SIMPGG, VINO

a. Purpose -- REGAIN is primarily a driver program to direct the recalculation of the cavity gain medium at the end of each iteration as shown by Figure 51. Subroutine REGAIN computer printouts follow Figure 51. The routine controls the type of kinetics calculation (numerical or analytical closed form), calculation of the FUHS effect on the medium density, generation of plots, and input/output of medium data on disk. Most of the control for this routine is read in from subroutine CAVITY.

b. Relevant formalism - The only formal calculations performed in REGAIN are the summation of cavity aerodynamics and FUHS effect induced optical path variations, and the averaging of newly calculated gain distribution with that of the previous iteration. A simple linear averaging or weighting algorithm is used:

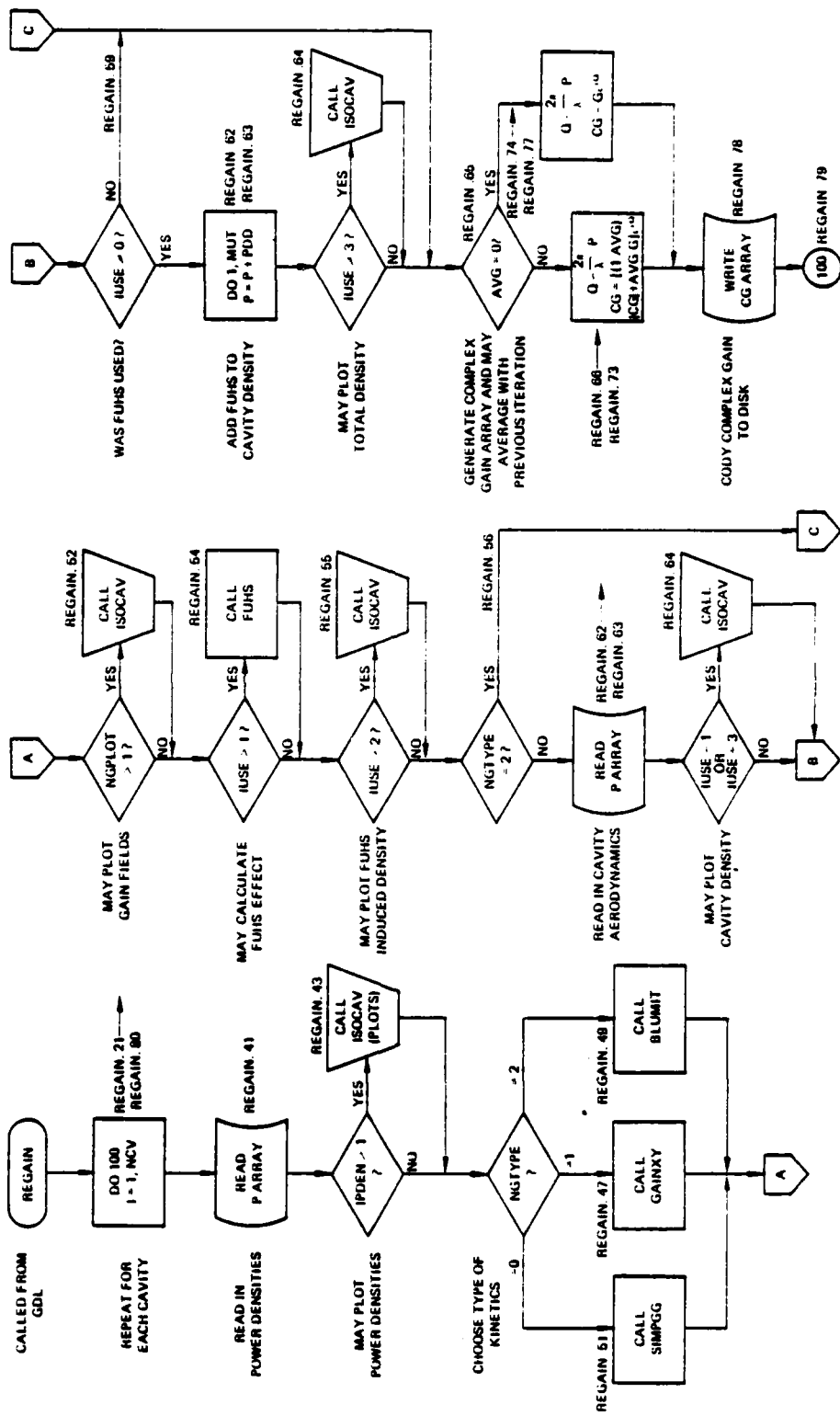


Figure 51. Subroutine REIGAIN flow chart.

$$G = (G_o (1-A) + G_c A) \exp \left((2\pi/\lambda) \text{OPD} \right) \quad (153)$$

where

G_o is the amplitude gain field from the previous iteration,

G_c is the newly calculated amplitude gain field,

OPD is the sum of optical path differences,

λ is the wavelength.

Argument List

NCT the number of cavity elements in the resonator

NIT the iteration number

Commons Modified

/CCG/

Variables Modified

CG the complex gain field

Relevant Variables

AVGG weighting factor for averaging new and old gain arrays -
defined by input to GDL

IBASE integer reference number to control reading and writing
power densities, gain, etc. to and from disk

IPDEN* flag for plotting power densities

IUSE* flag for FUHS calculation

NGPLOT* flag for plotting gain fields

NGTYPE* flag for controlling type of kinetics calculation

NSA* number of gain/phase segments

NXA* number of points in flow direction

NYA* number of points across cavity (side-to-side)

*Defined by input to CAVITY

```

SUBROUTINE REGAIN(NCT,NIT)
C THIS ROUTINE DIRECTS (1) THE RECALCULATION OF GAIN AFTER A
C RESONATOR ITERATION AND (2) THE GENERATION OF ANY SPECIFIED
C PLOTS OF THE CAVITY PARAMETERS.
LEVEL 2, CU,POU,G,P,CG
LEVEL 2, XC
COMMON/MLT/CU(16384),CFIL(16512),X(128),WL,NPTS,NPY,UMX,UMY
COMMON /CCG/ CG(17100)
COMMON /GGGGG/ G(17100)
COMMON/CAV2/ XC(5),YC(5),ZC(5),XA(5),YA(5),NS(5),XMC(5),YMC(5),
1 NGTY(5), NGPL(5), IU(5), IPU(5),
2 SSGAIN(190,5),SATIN(5),BETA(5),HMUS(5),
3 VEL(5),GAM(5),XMACH(5),TV1(5),TV2(5),TV3(5),TVN2(5),TSCAV(5),
4 PSCAV(5),PB(5),FN2(5),FCU2(5),FM20(5),FCU(5),FUZ(5),
5 TITLE(20),AVG(5),NSYM
DIMENSION POU(16384),P(16384),G(16384)
COMPLEX CU,CFIL,CG,CAKAY
EQUIVALENCE (POU(1),CFIL(1)) ,
X (P(1),CU(1))
CALL CPUTIM(ISRT)
C CAKAY = CMPLX(0.,2.*J.141592/WL)
TPIOL = 6.283184 / WL
DU 100 NCV=1,NCT
IBASE = 10*(NCV-1)*11
NGTYPE=NGTY(NCV)
NGPLUT=NGPL(NCV)
IUSE=IU(NCV)
IPUEN=IPU(NCV)
AVGG=AVG(NCV)
NSA=NS(NCV)
NXA=NX(NCV)
NYA=NY(NCV)/(NSYM*1)
MUT=NXA*NYA
NEWC = 0
MMM = 0
DU 90 L=1,NSA
IF ( NGPLOT.NE.-1 ) GO TO 18
NGPLUT = 3
IPDEN = 3
IUSE = 0
IF (IU(NCV).GE.1) IUSE=3
18 IPPP=IBASE*5*L
READ(IPP) (P(IZ),IZ=1,MUT)
REWIND IPPP
IF (IPDEN.GT.1) CALL ISOCAV(P, NCV, 2, L, NEWC,NIT,WL)
IF (IPDEN.EQ.1.OR.IPUEN.EQ.3) CALL VINO(P,NCV, L, NIT,2,MMM)
ICC=IBASE*L
C CALL NUMERICAL GAIN ROUTINE
IF (NGTYPE.EQ.1) CALL GAINXY(P,G,NCV,0)
C CALL MULTI-BEAM THERMAL BLOOMING ROUTINE
IF (NGTYPE.EQ.2) CALL BLUMIT(P,G,NCV,WL)
C CALL CLOSED FORM GAIN ROUTINE
IF (NGTYPE.EQ.0) CALL SIMPUG(P, G,NCV)
IF ( NGPLOT .GE.2) CALL ISUCAV(G,NCV,1, L,NEWC, NIT, WL),
IF (NGPLUT.EQ.1.OR.NGPLUT.EQ.3) CALL VINO(G,NCV, L, NIT,1,MMM)
IF (IUSE.GE.1) CALL FUMS(P,POU,NCV)
IF (IUSE.GE.2) CALL ISOCAV(POU, NCV, 3, L, NEWC, NIT, WL)
IF (NGTYPE.EQ.2) GO TO 25
HEAD (IBASE) (P(IZ),IZ=1,MUT)
REWIND IBASE
C
30 IF (IUSE.EQ.-1) GO TO 25
IF (IUSE.EQ.0.OR.IUSE.EQ.3) CALL ISUCAV(P,NCV,5,L,NEWC,NIT,WL)
IF (IUSE.EQ.0) GO TO 25
DU 22 J = 1,MUT
22 P (J) = P (J) + POU ( J )
IF ( IUSE .GE. 3 ) CALL ISUCAV(P,NCV,4,L,NEWC,NIT,WL)
25 IF (AVGG.EQ.0.) GO TO 21
HEAD(ICC) (CG(IZ),IZ=1,MUT)
REWIND ICC
REGAIN

```

```

REGAIN 2
REGAIN 3
REGAIN 4
REGAIN 5
REGAIN 6
CUMR2 9
REGAIN 7
CIUDENS 32
SQU77CY1 189
REGAIN 8
REGAIN 9
REGAIN 10
REGAIN 11
REGAIN 12
REGAIN 13
CIUDENS 33
REGAIN 15
SQU77CY1 190
CIUDENS 34
REGAIN 18
REGAIN 19
REGAIN 20
REGAIN 21
REGAIN 22
REGAIN 23
REGAIN 24
REGAIN 25
REGAIN 26
REGAIN 27
REGAIN 28
REGAIN 29
REGAIN 30
REGAIN 31
REGAIN 32
REGAIN 33
REGAIN 34
REGAIN 35
REGAIN 36
REGAIN 37
REGAIN 38
REGAIN 39
REGAIN 40
REGAIN 41
REGAIN 42
REGAIN 43
REGAIN 44
REGAIN 45
REGAIN 46
REGAIN 47
REGAIN 48
REGAIN 49
REGAIN 50
REGAIN 51
REGAIN 52
REGAIN 53
REGAIN 54
REGAIN 55
REGAIN 56
REGAIN 57
REGAIN 58
REGAIN 59
REGAIN 60
REGAIN 61
REGAIN 62
REGAIN 63
REGAIN 64
REGAIN 65
REGAIN 66
REGAIN 67

```

20	DO 110 I1=1,MUT	HEGAIN	68
	PHI = P(I1) * TPIUL	HEGAIN	69
C 110	CG(I1) = (G(I1)*(1.-AVGG)+CAUS(CG(I1))*AVGG) * CEXP(CAKAY*P(I1))	HEGAIN	70
110	CG(I1) = (G(I1)*(1.-AVGG)+CAUS(CG(I1))*AVGG) *	HEGAIN	71
	X CMPLX(COS(PHI) , SIN(PHI))	HEGAIN	72
	GO TO 23	HEGAIN	73
21	DO 112 I1=1,MUT	HEGAIN	74
	PHI = P(I1) * TPIUL	HEGAIN	75
C 112	CG(I1)=G(I1)*CEXP(CAKAY*P(I1))	HEGAIN	76
112	CG(I1)=G(I1)*CMPLX(COS(PHI) , SIN(PHI))	HEGAIN	77
23	WRITE(ICC) (CG(I2),I2=1,MUT)	HEGAIN	78
90	HE=INO ICC	HEGAIN	79
100	CONTINUE	HEGAIN	80
	WRITE(6,10)	HEGAIN	81
10	FORMAT(40M0GAIN HAS BEEN UPDATED FOR THE NEXT PASS)	HEGAIN	82
	IF(NGTYPE .EQ. 1) WRITE(6,11)	HEGAIN	83
11	FORMAT(31M USING NUMERICAL KINETICS MODEL)	HEGAIN	84
	IF(NGTYPE .EQ. 0) WRITE(6,12)	HEGAIN	85
	IF(NGTYPE .EQ. 2) WRITE(6,19)	HEGAIN	86
19	FORMAT(30M USING THERMAL BLOWING ANALYSIS)	HEGAIN	87
12	FORMAT(32M USING ANALYTICAL KINETICS MODEL)	HEGAIN	88
	IF(IUSE .GT. 0) WRITE(6,13)	HEGAIN	89
13	FORMAT(7UM0DENSITY VARIATIONS INDUCED BY LOWER LASER LEVEL RELAXAT	HEGAIN	90
	XION CALCULATED)	HEGAIN	91
	CALL CPULM(IFIN)	HEGAIN	92
	DELT=(ISHT-IFIN)/100.	HEGAIN	93
	WRITE(6,45) DELT	HEGAIN	94
45	FORMAT(1M0.G12.5+.49M SECONDS OF CPU TIME SPENT IN SUBROUTINE HEGAI	HEGAIN	95
	AN /1M1)	HEGAIN	96
	RETURN	HEGAIN	97
	END	HEGAIN	98

SUBROUTINE RGRD 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

	SUBROUTINE HNGD(NHGD)	HNGD	2
C		HNGD	3
C	THIS ROUTINE HEGRID CU FROM A NP15*02 ARRAY TO AN	HNGD	4
C	NHGD*02 ARRAY USING THE SAME GHID ELEMENT SIZE AS THE	HNGD	5
C	ORIGINAL ARRAY.	HNGD	6
C		HNGD	7
	LEVEL 2, CU,CFILM	HNGD	8
	COMMON/HELT/CU(16384),CFIL(16312),X(128),NL,NPTS,NPY,DMX,DMY	HNGD	9
	DIMENSION CFILM(32768)	HNGD	10
	COMPLEX CU,CFIL	HNGD	11
	EQUIVALENCE (CFIL(1),CFILM(1))	HNGD	12
	DX=X(2)-X(1)	HNGD	13
	NFAC = NPTS/NPY	HNGD	14
	NYAD=(NRGD-NPTS)/2	HNGD	15
	NXAD=(NRGD-NPTS)/2	HNGD	16
	X(1)=DX*(1-NHGD)/2.	HNGD	17
	DO 10 I=2,NHGD	HNGD	18
	X(I)=X(I-1)+DX	HNGD	19
10	CONTINUE	HNGD	20
C	WRITE(6,101)X(1),X(NHGD)	HNGD	21
C 101	FORMAT(//10X,6MX(1) =.G12.4+.5X,9MX(NHGD) =.G12.4//)	HNGD	22
C	CALL ZERO (CFIL(1),CFIL(16384))	HNGD	23
	DO 173 IZENO=1,32768	HNGD	24
173	CFILM(IZENO)=0.	HNGD	25
	DO 20 J=1,NPY	HNGD	26
	INX=NRGD*(NYAD+J-1)+NXAD	HNGD	27
	NBASE=(J-1)*NPTS	HNGD	28
	DO 30 I=1,NPTS	HNGD	29
	CFIL(INOX+I)=CU(NBASE+I)	HNGD	30
30	CONTINUE	HNGD	31
20	CONTINUE	HNGD	32
	NPTS=NRGD	HNGD	33
	NPY = NPTS/NFAC	HNGD	34
	NSQR=NPTS*NPY	HNGD	35
	DO 40 IM=1,NSQR	HNGD	36
	CU(IM)=CFIL(IM)	HNGD	37
40	CONTINUE	HNGD	38
	RETURN	HNGD	39
	END	HNGD	40

25. SUBROUTINE RGRD

This routine regrid a complex amplitude field by adding zeroes to the array on all sides of the input field. Figure 52 is the flow chart for subroutine RGRD. Points added have the same separation as the original field. No interpolation or other formal calculation is necessary. Use of this routine has the effect of increasing the guard band around the field.

Argument List

NRGD desired number of grid points across field

Relevant Variables

DX separation of grid points before and after regridding
INDX counter or index used to locate old grid within the new grid
NSQR total number of points in regrided field

Commons Modified

/MELT/

Variables Modified

CFIL temporary field storage array
CU complex amplitude field
NPTS number of grid points in x-dimension
NPY number of grid points in y-dimension
X x-position array

26. SUBROUTINE ROSN

a. Purpose -- The purpose of subroutine ROSN is to provide an accurate and rapid numerical interpolation subprogram for the evaluation of cavity-induced density perturbations. The subroutine uses cubic spline processed data representing aerodynamically parameterized $\frac{\Delta \rho}{\rho}$ data to interpolate to the cavity mesh for the run in question as shown in the ROSN subroutine flow chart (Fig. 53). Subroutine ROSN requires that the user specify the relevant cubic spline coefficients and $\frac{\Delta \rho}{\rho}$ values. The subroutine calculates $\Delta \phi$ for an arbitrary cavity mesh point, (x,y).

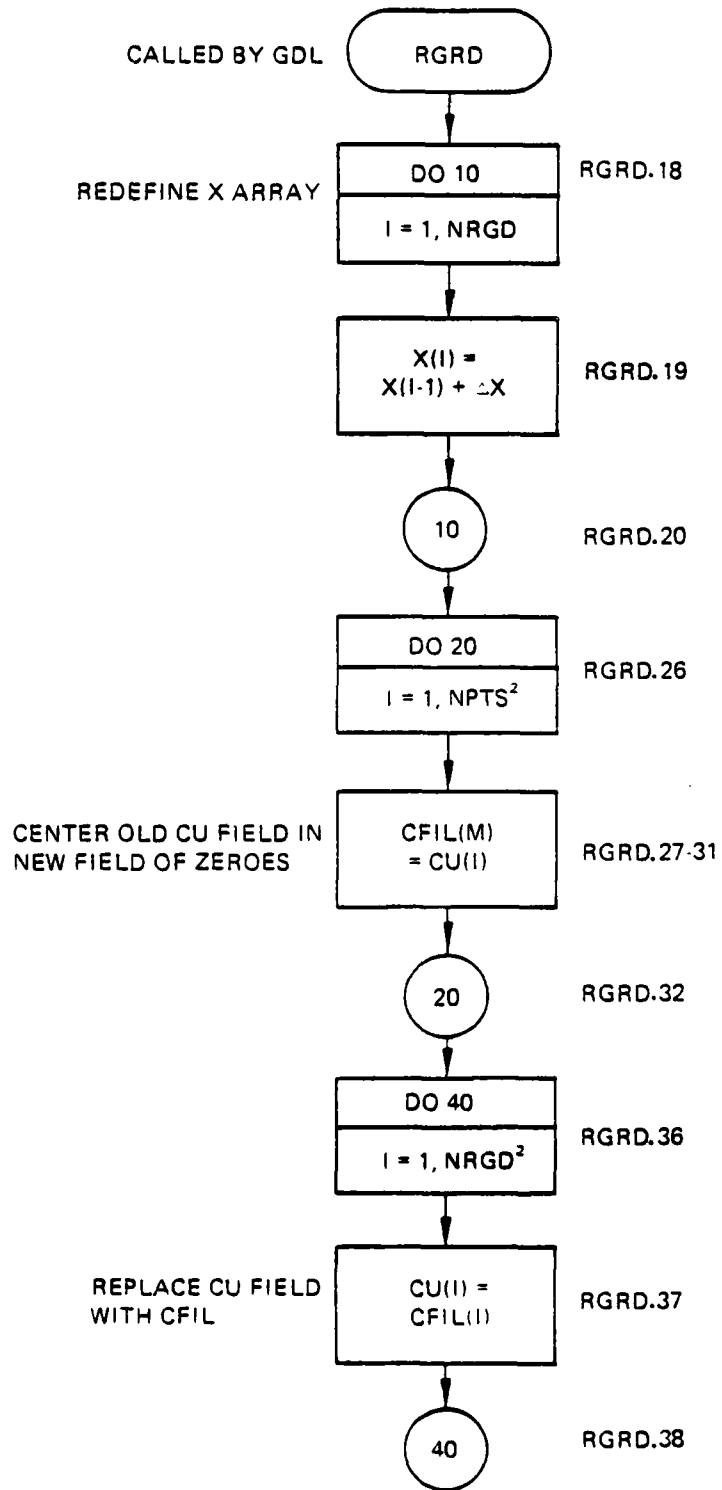


Figure 52. Subroutine RGRD flow chart.

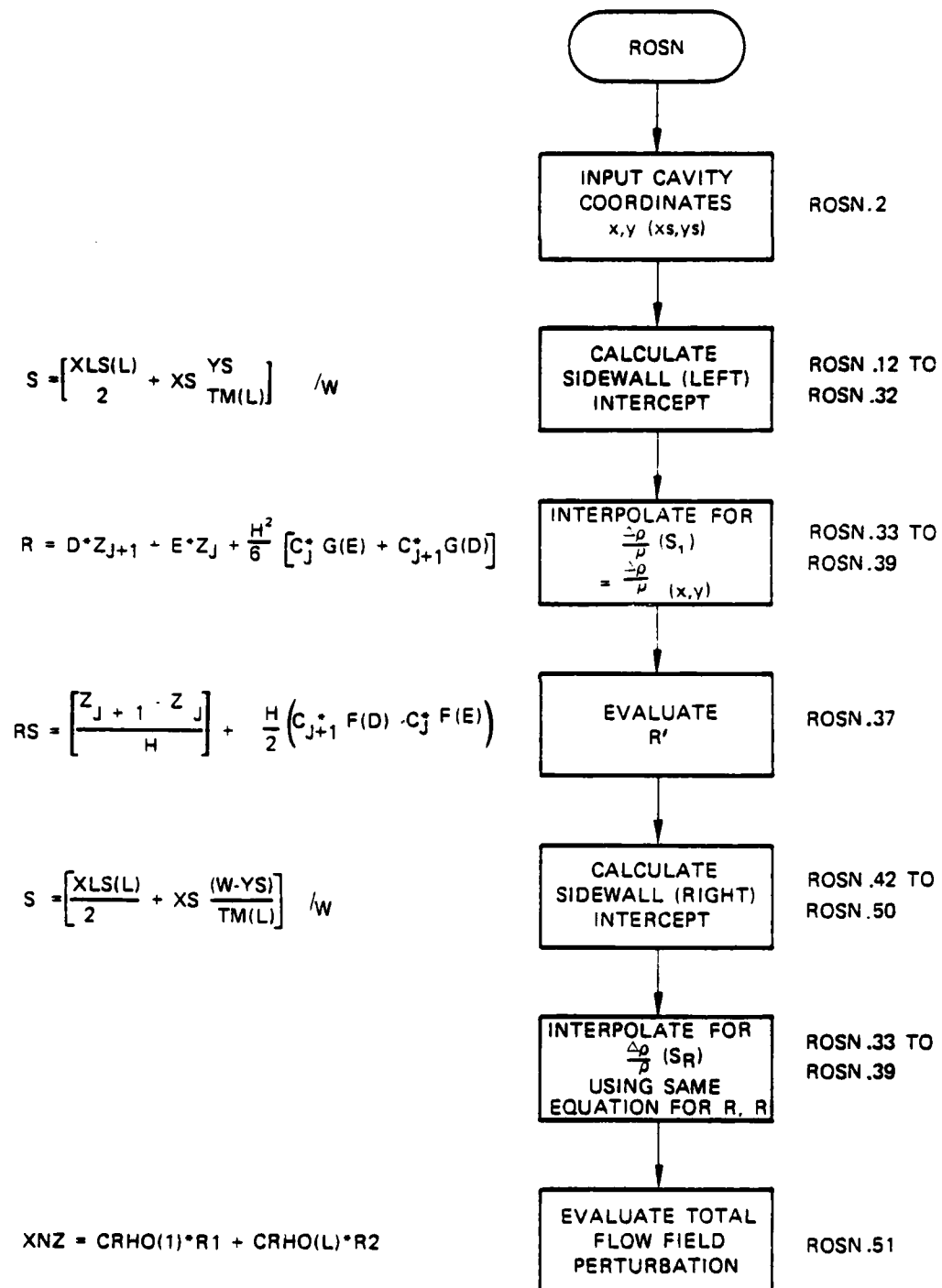


Figure 53. Subroutine ROSN flow chart.

b. Relevant formalism -- The SOQ Cavity coordinate system represents a regular mesh upon which many perturbations are applied. High Mach number flow produces ordered density gradients which may degrade beam phase relationships. Given arbitrary flow field interferometry it is possible to parameterize fringe shift ($\frac{\Delta\rho}{\rho}$ or ΔOPD) as a function of sidewall parameter s , where s is determined from the cavity sidewall projection of Mach lines, as shown in Figure 54.

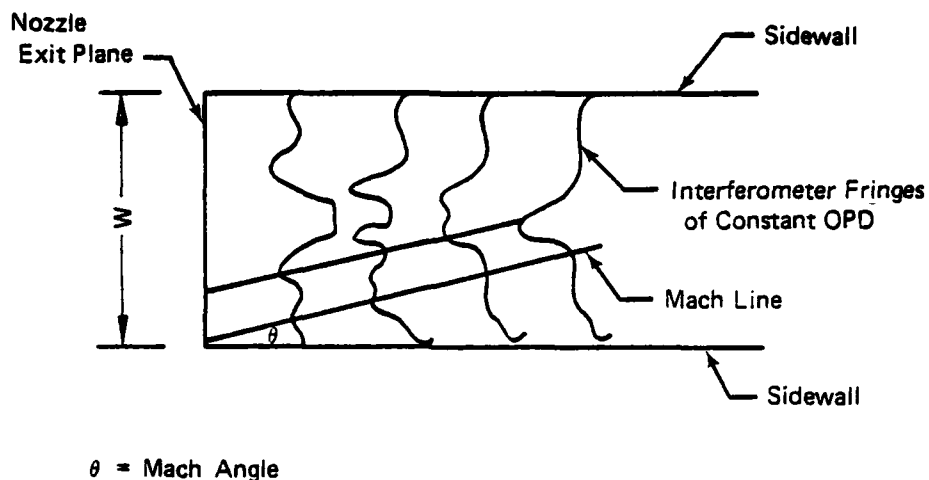


Figure 54. Fringe shift as a function of sidewall parameter.

From interferometry data and the above concept of sidewall projected data, the following parametric curves may be defined:

The curves shown in Figure 55 are fit using cubic splines, and the table or arrays of $\frac{\Delta\rho}{\rho} = f(s^*)$ and $C = g(s^*)$ (spline coeff) are stored in program DENSY. Subroutine ROSN is used to interpolate from (x,y) in the cavity to equivalent sidewall position s_{right} and s_{left} to determine using the above spline coefficients, an interpolated value of $\frac{\Delta\rho}{\rho} \Big|_{\text{left}} = f(s_{\text{left}}) = H(x,y)$

$$\frac{\Delta\rho}{\rho} \Big|_{\text{right}} = g(s_{\text{right}}) = K(x,y) \quad (154)$$

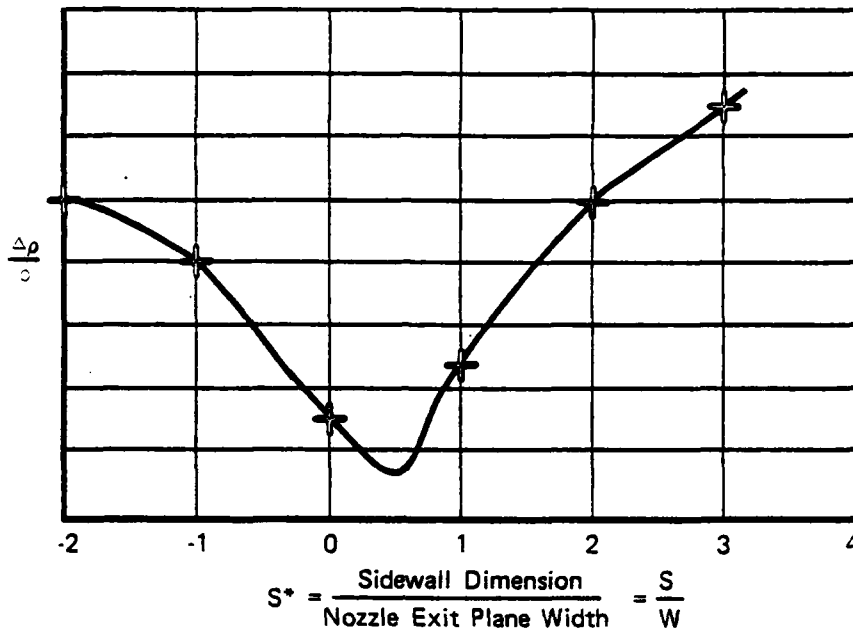


Figure 55. Parametric curves of Mach lines.

The $\left(\frac{\Delta\rho}{\rho}\right)$ at the point (x,y) is given, from supersonic flow theory as:

$$\left.\frac{\Delta\rho}{\rho_{CL}}\right|_{\text{Total}} = \left.\frac{\Delta\rho}{\rho_{CL}}\right|_{\text{left}} + \left.\frac{\Delta\rho}{\rho_{CL}}\right|_{\text{right}}$$

$$\Delta\phi = \frac{2\pi}{\lambda} C \left.\frac{\Delta\rho}{\rho_{CL}}\right|_{\text{Total}} \rho_{CL}$$

$$\Delta\phi = \Delta\phi(x,y)$$

$$\left(\frac{\Delta\rho}{\rho}\right) = \frac{\Delta\rho(x,y)}{\rho}$$

The Spline interpolator is:

	SUBROUTINE ROSN(AS,YS,XNZ)	HUSN	2
C	CAVITY DENSITY FIELD INTERPOLATION ROUTINE	HUSN	3
C	THIS ROUTINE USES SPLINE COEFFICIENTS TO INTERPOLATE THE CAVITY	HUSN	4
C	DENSITY FIELD (DELTA RHO/RHO AND SPLINE COEFFICIENT VERSUS	HUSN	5
C	SIDEWALL PARAMETERS) ONTO THE CAVITY MESH.	HUSN	6
	COMMON/LENSY/Y(51,2),Z(51,2),C(51,2),TM(2),XLS(2),w	HUSN	7
X	AMULT(2), CHMO(2), M(2), TITLE(20), LL	HUSN	8
	DATA J/2/	HUSN	9
	F(A)=A*A-1./3.	HUSN	10
	G(A)=A*(A*A-1.)	HUSN	11
	L = 1	HUSN	12
	KY=M(L) -1	HUSN	13
	MM = M(L)	HUSN	14
	ITEST=0	HUSN	15
	S=(XLS(L)/2.+XS-YS/TM(L))/w	HUSN	16
6	IF(S=Y(1,L))30,7,7	HUSN	17
7	IF(S=Y(MM,L))8,8,30	HUSN	18
8	IF(J-KY)20,20,9	HUSN	19
9	J=KY	HUSN	20
20	YD1=Y(J,L)-S	HUSN	21
	YD2=Y(J+1,L)-S	HUSN	22
	IF(YD1*YD2)5,5,22	HUSN	23
22	IF(YD1)10,10,23	HUSN	24
10	J=J+1	HUSN	25
	IF(J-KY)20,11,11	HUSN	26
11	J=KY	HUSN	27
	GO TO 5	HUSN	28
23	J=J-1	HUSN	29
	IF(J)12,12,20	HUSN	30
12	J=1	HUSN	31
5	JP=J+1	HUSN	32
	H=Y(JP,L)-Y(J,L)	HUSN	33
	D=(S-Y(J,L))/H	HUSN	34
	E=1.-D	HUSN	35
	H=D*Z(JP,L)+E*Z(J,L)+H*M/6.*(C(J,L)*G(E)+C(JP,L)*G(D))	HUSN	36
	RS=(Z(JP,L)-Z(J,L))/H+H/2.*(C(JP,L)*F(D)-C(J,L)*F(E))	HUSN	37
	GO TO 31	HUSN	38
30	R=0.	HUSN	39
	HS=0.	HUSN	40
31	IF(ITEST)32,32,33	HUSN	41
32	ITEST=1	HUSN	42
	R1=R	HUSN	43
	L = LL	HUSN	44
	MM = M(L)	HUSN	45
	KY = MM - 1	HUSN	46
	RS1=RS	HUSN	47
	S=(XLS(L)/2.+XS-(W-YS)/TM(L))/W	HUSN	48
	J=MM-J	HUSN	49
	GO TO 6	HUSN	50
33	XNZ=CHMO(1) * R1 * CHMO(L) * H	HUSN	51
	RETURN	HUSN	52
	END	HUSN	53

27. SUBROUTINE LINTERP

a. Purpose -- This subroutine is used within the SOQ code to linearly interpolate sidewall projected $\frac{\Delta\rho}{\rho}$ cavity density information from sidewall projection to the cavity mesh. Data $\frac{\Delta\rho}{\rho}$ are stored in compressed form as univariate curves of $\frac{\Delta\rho}{\rho}$ versus sidewall projection parameters s, from which $\frac{\Delta\rho}{\rho}$ at any point in the GDL cavity may be obtained as shown in Figure 56.

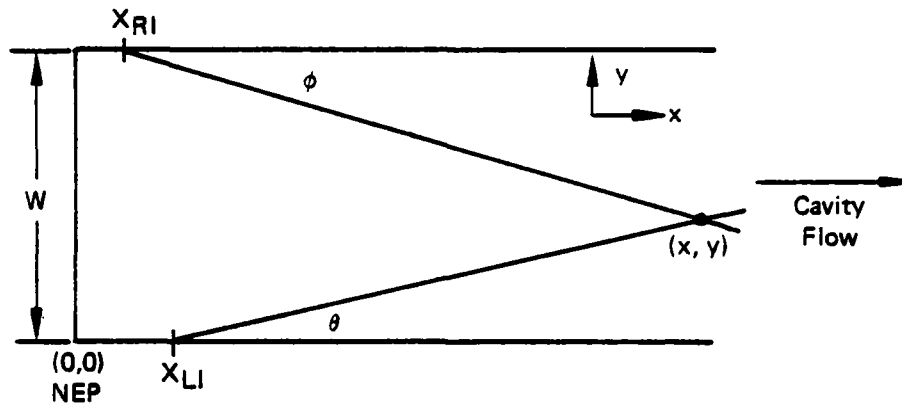


Figure 56. $\Delta\rho/\rho$ cavity density information.

The interpolated $\frac{\Delta\rho}{\rho}$ value is calculated to determine the equivalent flow-induced lens which is to be applied to the propagating wavefront. The lens is the result of flow-induced inhomogenities such as ordered density gradients (weak shocks) and uneven thermal distribution.

The LINTERP subprogram (Fig. 57) calculates the sidewall parameters from interpolated cavity position (x,y) and Mach angle. With "s" determined for both right and left cavity sidewall projections a $\frac{\Delta\rho}{\rho}$ contribution can be determined for both sidewalls and linearly combined to give $(\frac{\Delta\rho}{\rho})_{TOTAL} = f(x,y)$.

b. Relevant formalism

Left Intercept:

$$\tan\theta = \frac{y}{(x-x_{LI})} \quad x_{LI} = -\frac{y}{\tan\theta} + x \quad (156)$$

where

(x,y) = interpolate position
 x_{LI} = Left intercept
 $\tan\theta$ = tangent of Mach angle

sidewall parameter s

$$s_L = \frac{x_{LI}}{W} = \frac{(x-y/\tan\theta)}{W}$$

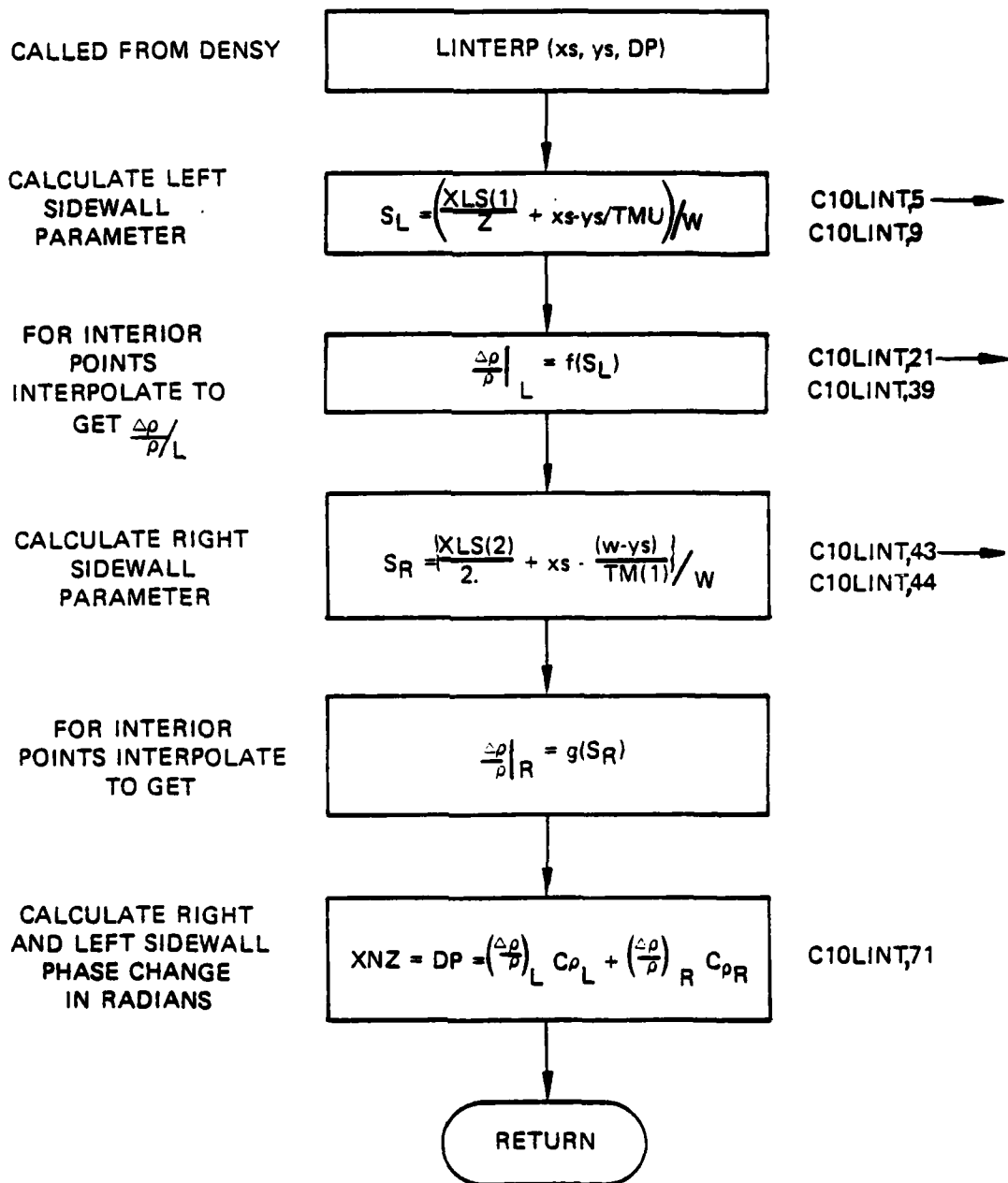


Figure 57. Subroutine LINTERP organization.

Right Intercept:

$$\tan\theta = \frac{w-y}{x-x_{R_I}} \quad (157)$$

$$(x - x_{R_I}) \tan\theta = (w-y) - x \tan\theta$$

$$x_{R_I} = \frac{-(w-y)}{\tan\theta} + x$$

$$S_R = \frac{x_{R_I}}{W} = \frac{x-(w-y)/\tan\theta}{W} \quad (158)$$

where

w = cavity width

tan θ = tangent Mach angle

(θ = positive angle)

Commons modified

NONE

Definition of relevant variables

TM Tangent of Mach angle

XLS Arbitrary sidewall intercept offset (cm)

w Cavity width (cm)

CRHO Composite constant = $\frac{2\pi}{\lambda} \text{CAL} \rho_0$

Subroutine LINTERP computer printouts follow.

SUBROUTINE LINTERP 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE LINTERP(XS,YS,ANZ)
COMMON/LENSY/Y(51,2),Z(51,2),C(51,2),TM(2),XLS(2),W
X MULT(2),CHMU(2),M(2),FILE(20),LL
C***** CALCULATE SIDEWALL PARAMETER *****
L=1
MM=M(L)
SL=(ALS(L)/2. * XS -YS/TM(L))/W
IF(SL.LT.Y(1,L)) GO TO 5
IF(SL.GE.Y(MM,L))GO TO 6
C ***** FIND S POSITION IN Y ANRAY*****
DO 10 I = 1,MM
IF(SL.GT.Y(I,L)) GO TO 10
KL=I
KLM=I-1
YSL=Y(I,L)
YSLM=Y(I-1,L)
GO TO 15
10 CONTINUE
15 CONTINUE
C I O L I N T 1
C I O L I N T 2
C I O L I N T 3
C I O L I N T 4
C I O L I N T 5
C I O L I N T 6
C I O L I N T 7
C I O L I N T 8
C I O L I N T 9
C I O L I N T 10
C I O L I N T 11
C I O L I N T 12
C I O L I N T 13
C I O L I N T 14
C I O L I N T 15
C I O L I N T 16
C I O L I N T 17
C I O L I N T 18
C I O L I N T 19

```

C *****DETERMINE UHMU QVM MHOCL *****	C1OLINT	20
C ***** FOR INTERIOR POINTS *****	C1OLINT	21
YU1L=YSL - YSLM1	C1OLINT	22
YD2L=SL - YSLM1	C1OLINT	23
DMMO1= Z(KL,L) -Z(KLM1,L)	C1OLINT	24
DMMO2= Z(KLM1,L)	C1OLINT	25
LLL=1	C1OLINT	26
C IF(XS.GT.20.)	C1OLINT	27
C XWRITE(6,92)KL,KLM1,Y(KL,L),Y(KLM1,L),Z(KL,L),Z(KLM1,L)	C1OLINT	28
92 FORMAT(5X,0U,10 LOOP,2I5,4(5X,E15.7))	C1OLINT	29
DMMOL=(YD2L/YU1L)*DMMO1 + DMMO2	C1OLINT	30
GO TO 20	C1OLINT	31
5 DMMOL = Z(1,L)	C1OLINT	32
LLL=2	C1OLINT	33
GO TO 20	C1OLINT	34
6 DMMOL = Z(MM,L)	C1OLINT	35
LLL=3	C1OLINT	36
20 CONTINUE	C1OLINT	37
C IF(XS.GT.20.)WRITE(6,99)LLL,SL,DMMOL	C1OLINT	38
99 FORMAT(10X,I5,2(5X,E15.7),* LLL SL DMMOL,*)	C1OLINT	39
C***** CALCULATE SIDEWALL PARAMETER (RIGHT)*****	C1OLINT	40
L=LL	C1OLINT	41
MM= M(L)	C1OLINT	42
SH=(XLS(L)/2. + XS -(W-Y5)/TM(L))/W	C1OLINT	43
IF(SH .LT. Y(1,L))GO TO 7	C1OLINT	44
IF(SH .GE. Y(MM,L))GO TO 4	C1OLINT	45
DO 40 I=1,MM	C1OLINT	46
IF(SH.GT. Y(I,L)) GO TO 40	C1OLINT	47
KHI=I	C1OLINT	48
KHM1= I - 1	C1OLINT	49
YDH1=Y(KH,L) - Y(KHM1,L)	C1OLINT	50
YDH2= SH - Y(KHM1,L)	C1OLINT	51
GO TO 45	C1OLINT	52
40 CONTINUE	C1OLINT	53
45 CONTINUE	C1OLINT	54
DMMOR1= Z(KH,L) -Z(KHM1,L)	C1OLINT	55
DMMOR2= Z(KHM1,L)	C1OLINT	56
DMMOH=(YDH2/YDH1)*DMMO1 + DMMO2	C1OLINT	57
KKK= 1	C1OLINT	58
C IF(XS.GT.20.)	C1OLINT	59
C XWRITE(6,93)KH,KHM1,Y(KH,L),Y(KHM1,L),Z(KH,L),Z(KHM1,L)	C1OLINT	60
93 FORMAT(5X,0U,10 LOOP,2I5,4(5X,E15.7))	C1OLINT	61
GO TO 50	C1OLINT	62
7 DMMOH = Z(1,L)	C1OLINT	63
KKK=2	C1OLINT	64
GO TO 50	C1OLINT	65
8 DMMOH = Z(MM,L)	C1OLINT	66
KKK=3	C1OLINT	67
50 CONTINUE	C1OLINT	68
C IF(XS.GT.20.)WRITE(6,199)KKK,SH,DMMOH	C1OLINT	69
199 FORMAT(10X,I5,2(5X,E15.7),* KKK,SH,DMMOH,*)	C1OLINT	70
XNZ= DMMOL*CHMO(1) + DMMOH*CHMU(L)	C1OLINT	71
C IF(XS.GT.20.)WRITE(6,249) CHMU(1),CHMU(L)	C1OLINT	72
249 FORMAT(20X,0CRMU(1),CRMU(L) *,2(E15.7),*)	C1OLINT	73
RETURN	C1OLINT	74
END	C1OLINT	75

28. SUBROUTINE ROSN6

a. Purpose -- Subroutine ROSN6 (flow chart organization shown in Fig. 58) is incorporated into the SOQ code to allow inclusion of the cavity density field from direct interferogram data reduction. The data from interferometry are assumed to have been fit in the y (parallel to NEP) direction by cubic splines, using spaced points (not necessarily equal).

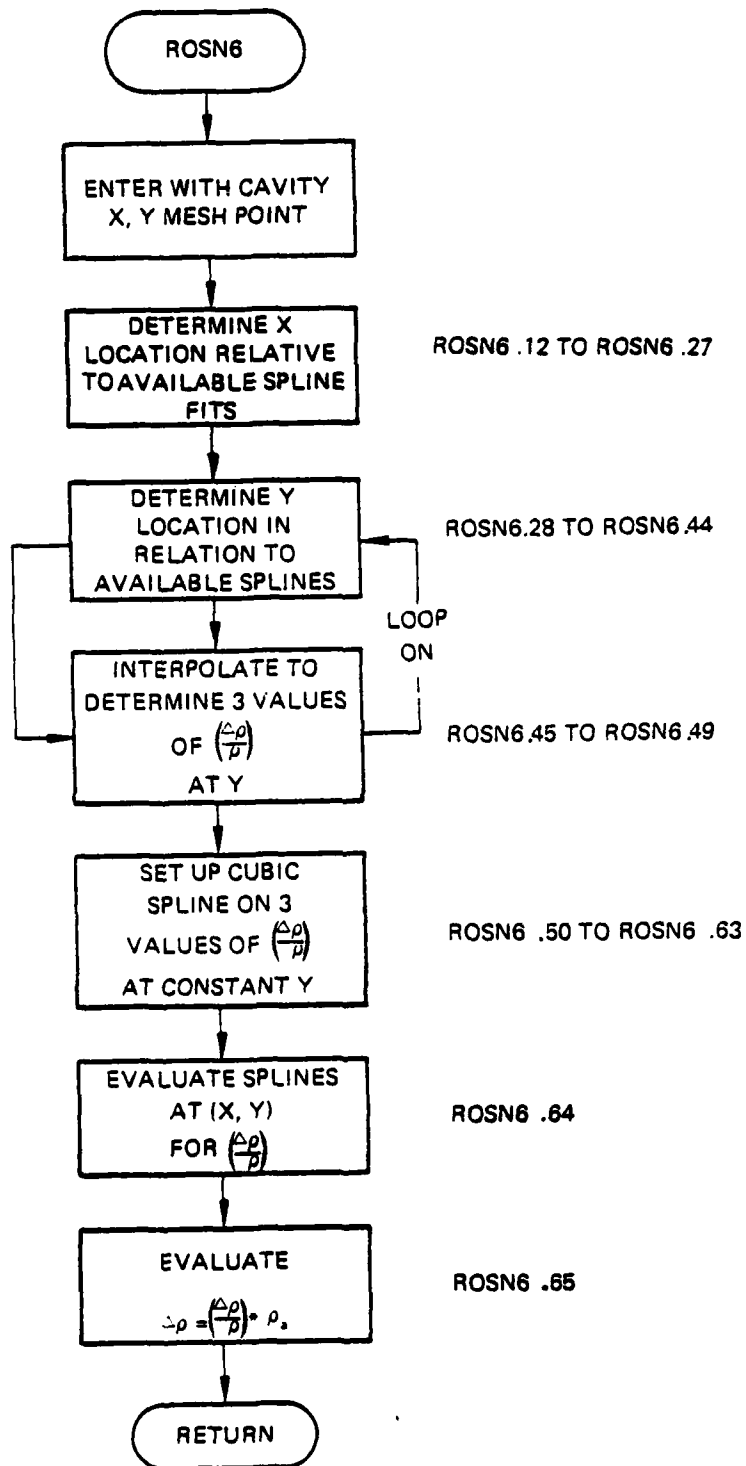


Figure 58. Subroutine ROSN6 organization.

Subroutine ROSN6 is a bivariate interpolation of the spline fit data using cubic splines.

b. Relevant formalism -- Subroutine ROSN6 uses the following procedure to interpolate the available spline data for an arbitrary cavity mesh point, (x,y) , shown in Figure 59.

- (1) Locate * in the spline fit data.
- (2) Interpolate, using the spline fits at constant y , for the value of $\frac{\Delta\rho}{\rho}$ at the nearest three x values, (Δ) .
- (3) Construct a cubic spline in the direction (x_i, y^*) and evaluate at (x^*, y^*)
- (4) Modify $\frac{\Delta\rho}{\rho_{CL}}(x^*, y^*)$ by $\frac{\Delta\rho}{\rho_{CL}}(x^*, y^*)$ to obtain $\Delta\rho$ in the desired units.

See page 214 for subroutine ROSN6 computer printouts.

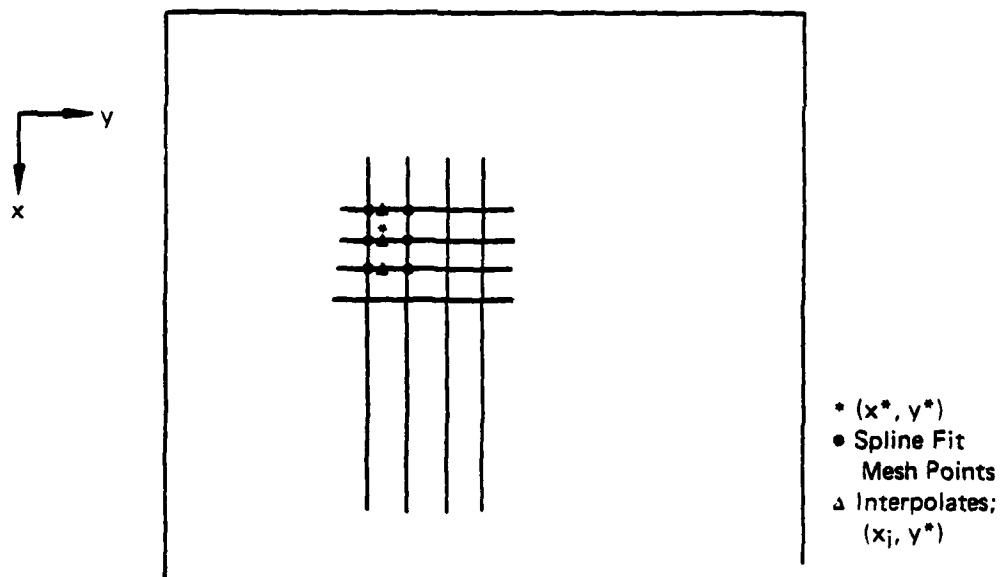


Figure 59. Available spline data for an arbitrary cavity mesh point.

Commons modified

/MELT/ not modified

/MELT/ is used to transfer in the following data:

x<=>cavity flow direction coordinates of spline fit data

y<=>orthogonal coordinates of spline coefficients

z<=>ordinate at each (x_i, y_j)

C<=>corresponding spline coefficients

M<=>Index array for constant x.

N

ROCL intended to be ρ at the center line but may be an arbitrary scaling parameter.

Relevant Variables

xx cavity x-position

yy cavity y-position

XNZ ordinate interpolated at (x,y) , normally $\Delta\rho = f(x,y)$

SUBROUTINE ROSN6 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.17

```

SUBROUTINE ROSN6 (XX,YY,XNZ)
C THIS ROUTINE IS USED TO INTERPOLATE THE CAVITY DENSITY FIELD
C (DELTA RHO/RHO AND SPLINE COEFFICIENT VERSUS X AND Y) ONTO THE
C CAVITY MESH.
LEVEL 2. PDUM
COMMON / MELT / PDUM(20000), X(21),
X Y(21,81), Z(21,81), C(21,81), M(21), N, ROCL, OUMYS(40778)
DIMENSION F(3), FM(3)
DATA II,J/2,2/
G(A)=A*(A-1.)
C COMPUTE LOCATION OF XX IN X(I)    X(I) .LE. XX .LE. X(N)
KX=N-2
10 X01=X(II)-XX
X02=X((II+1))-XX
IF(X01*X02)2,2,12
12 IF(X01 .GT. 0.) GO TO 13
II = II+1
IF(II .LT. KX) GO TO 10
II=KX
GO TO 2
13 II = II-1
IF(II .GT. 0) GO TO 10
II = 1
C COMPUTE THREE VALUES OF Z AND UZ/UY AT YY
2 L=II+2
KK=0
C COMPUTE LOCATION OF YY IN Y(M(I))    Y(I) .LE. YY .LE. Y(M(I))
DO 6 I=II,L
KK=KK+1
KY=M(I)-1
IF(J .GT. KY) J=KY

```

RUSN6	2
RUSN6	3
RUSN6	4
RUSN6	5
RUSN6	6
RUSN6	7
COMR2	10
RUSN6	9
RUSN6	10
RUSN6	11
RUSN6	12
RUSN6	13
RUSN6	14
RUSN6	15
RUSN6	16
RUSN6	17
RUSN6	18
RUSN6	19
RUSN6	20
RUSN6	21
RUSN6	22
RUSN6	23
RUSN6	24
RUSN6	25
RUSN6	26
RUSN6	27
RUSN6	28
RUSN6	29
RUSN6	30
RUSN6	31
RUSN6	32

20	YU1=Y(I,J)-YY	HUSN6	33
	YD2 = Y(I,J+1)-YY	HUSN6	34
	IF(YU1*YU2)5.5.22	HUSN6	35
22	IF(YU1 .GT. 0.) GO TO 23	HUSN6	36
	J=J+1	HUSN6	37
	IF(J .LT. KY) GO TO 20	HUSN6	38
	J=KY	HUSN6	39
	GO TO 5	HUSN6	40
23	J=J-1	HUSN6	41
	IF(J .GT. 0) GO TO 20	HUSN6	42
	J=1	HUSN6	43
5	JP=J+1	HUSN6	44
	M=Y(I,JP)-Y(I,J)	HUSN6	45
	D=(YY-Y(I,J))/M	HUSN6	46
	E=1.-D	HUSN6	47
	F(KK)=D*Z(I,JP)+E*Z(I,J)+M*M/b.*(C(I,J)*G(E)+C(I,JP)*G(D))	HUSN6	48
6	CONTINUE	HUSN6	49
C	COMPUTE Z, DZ/DX, UZ/DY AT XX FROM CUBIC SPLINE THROUGH F AND FP	HUSN6	50
	M1=X(II+1)-X(II)	HUSN6	51
	M2=X(II+2)-X(II+1)	HUSN6	52
	IF(X(II+1)-XX)7.8.0	HUSN6	53
7	U=(XX-X(II+1))/M2	HUSN6	54
	K=2	HUSN6	55
	H=M2	HUSN6	56
	GO TO 9	HUSN6	57
8	U=(XX-X(II))/M1	HUSN6	58
	K=1	HUSN6	59
	H=M1	HUSN6	60
9	E=1.-D	HUSN6	61
	CU=2.*((F(J)-F(2))/M2-(F(2)-F(1))/M1)/(M1*M2)	HUSN6	62
	TEM=M*M/b.*(G(E)+G(U))	HUSN6	63
	XN=U*F(K+1)+E*F(K)+CU*TEM	HUSN6	64
	XNZ=RUCL*XN	HUSN6	65
	RETURN	HUSN6	66
	END	HUSN6	67

29. SUBROUTINE SIMPGG

a. Purpose -- SIMPGG is used to calculate loaded gain for GDL cavities. It uses the E. A. Sziklas closed-form gain solution as derived in Reference 1, instead of numerically solving the appropriate GDL kinetics differential equations. SIMPGG also finds the intensity emitted at the gain/phase segment for use in FUHS. Figure 60 shows the SIMPGG organization.

b. Relative formalism -- The effect of the interaction of the light with the medium results in an amplification of the light beam as well as a phase change. Analytically this effect on the field is written

$$U(x,y) = t(x,y)U(x,y)$$

(159)

with

$$t(x,y) = e^{\frac{1}{2}g(x,y)\Delta L} e^{-i\frac{2\pi}{\lambda}\Delta n\Delta L}$$

AD-A103 285

UNITED TECHNOLOGIES CORP WEST PALM BEACH FLA
SYSTEM OPTICAL QUALITY USERS GUIDE. PART 2.(U)
MAR 80 J L FORGHAM, S S TOWNSEND

F/G 20/5

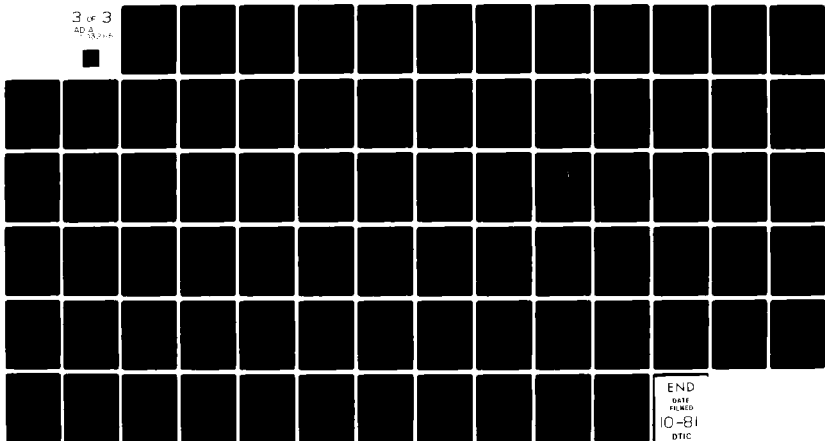
F29601-77-C-0025

UNCLASSIFIED

AFWL-TR-79-141-PT-2

NL

3 of 3
AD A103 285



END
DATE
FILMED
10-81
DTIC

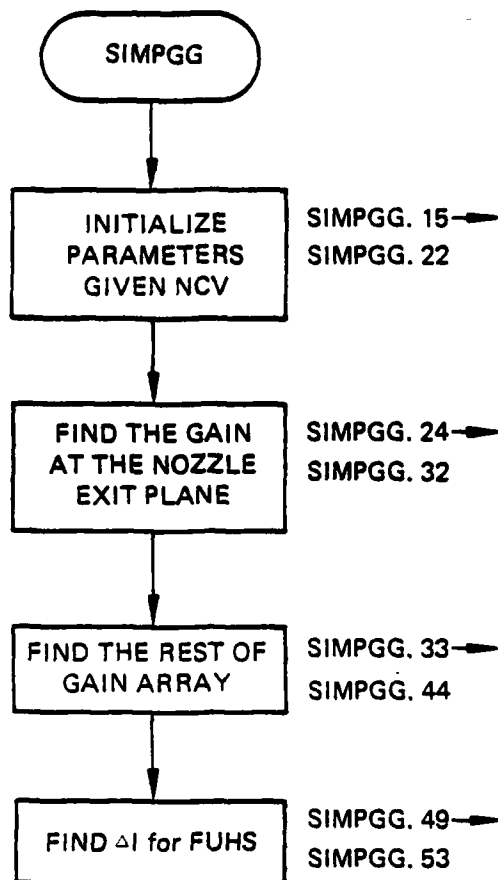


Figure 60. Subroutine SIMPGG organization.

ΔL is width of the medium under consideration, $g(x,y)$ is the loaded gain coefficients and $\Delta n(x,y)$ is change in index of refraction due to density variations.

The factor of 1/2 in the exponent is due to the fact that gain is intensity, not amplitude, related:

$$I_{OUT} = I_{IN} e^{g\Delta L} = GI_{IN} \quad (160)$$

where

$$I = |u|^2$$

SIMPGG determines $g(x,y)$ analytically using expression

$$g(x,y) = \left[\frac{g_0(x,y)}{1 + I(x,y)/I_{SAT}} \right] e^{\left(\frac{-X_{CO_2} \beta}{X_{N_2} V} \right) \int_{x_0}^x dx} \frac{I(x,y)}{I_{SAT} + I(x,y)} \quad (161)$$

and using the trapezoidal rule for the integral, where $g_0(x,y)$ is the small-signal gain coefficient found in subroutine GAINXY.

Note that

$$g(x,y) \Big|_{I(x,y) = 0} = g_0(x,y) \quad (162)$$

I_{sat} is the "saturation intensity"

$$I_{SAT} = \frac{h\nu\beta}{\sigma} \quad (163)$$

where

$h\nu$ is the photon energy, β the lower laser level relaxation rate, and σ the optical cross section for the transition. I_{sat} is also defined in subroutine GAINXY.

Where the FUHS routine is to be called to calculate heat increase in the gas due to lower level decay, the intensity change in the beam is needed for each gain phase segment, thus giving the heat release.

Consider Figure 61 of a gain/phase segment

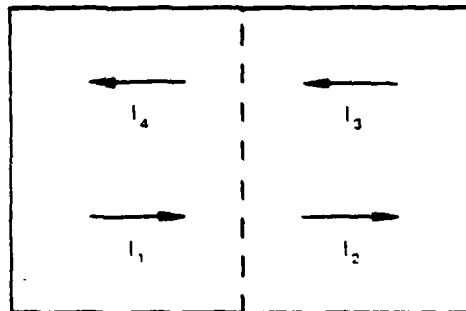


Figure 61. Gain/phase segment.

Then for each (I,J)

$$\Delta I = (I_1 + I_3) - (I_2 + I_4) \quad (164)$$

the quantity stored in the array PPD after a complete round trip is the average of the right running wave $(I_1 + I_2)/2$ plus the average of the left running wave $(I_3 + I_4)/2$.

Therefore

$$PPD = (I_1 + I_2 + I_3 + I_4)/2 \quad (165)$$

but $I_2 = GI$, and $I_4 = GI_3$

so $\Delta I = (1-G) (I_1 + I_3)$

and $PPD = \left(\frac{1+G}{2}\right)(I_1 + I_3)$

therefore

$$\Delta I = 2 \left(\frac{1-G}{1+G}\right) * PPD \quad (166)$$

Knowing the total power change due to ΔI and the quantum efficiency η , the total heat released is found. The factor $\frac{1}{\Delta z} \left(\frac{1-\eta}{\eta}\right)$ is discussed in FUHS.

c. Fortran

Argument List

PPD = Total intensity (left running + right running waves) --

Becomes $\frac{1}{\Delta z} \left(\frac{1-\eta}{\eta}\right) \Delta I$ for use in FUHS

GG = Gain = $e^{-g\Delta z/2}$

NCV = cavity number

Commons modified -- none

Subroutines called - none.

Subroutine SIMPGG computer printouts follow.

SUBROUTINE SIMPGG 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE SIMPGG (PPD,GG,NCV)
C CLOSED FORM GAIN ALGORITHM
C THIS ROUTINE USES THE E.A.SZIRLAS CLOSED FORM GAIN SOLUTION FOR
C CU2 TO CALCULATE LOADED GAIN FOR THE GUL CAVITIES.
LEVEL 2, XC,PPD,GG
COMMON/CAV2/ XC(5),YC(5),ZC(5),NX(5),NY(5),NS(5),XMC(5),YMC(5),
2 NGTYP(10), IUS(10), SSGAIN(190,5),SATIN(5),BETA(5),NMUS(5),
3 VEL(5),GAM(5),XMACH(5),TV1(5),TV2(5),TV3(5),TVN2(5),TSCAV(5),
4 PSCAV(5),PB(5),FN2(5),FCU2(5),FM2U(5),FCU(5),FU2(5),
5 TITLE(20), AVG(5),NSYM
DIMENSION GG(1), PPD( 16384),
2 G(190),SGAINX(190),WINTS(190)
C CALL CPUFIN(ISRT)
NSA=NS(NCV)
NYA=NY(NCV) / (NSYM*1)
NXA=NX(NCV)
SAT=SATIN(NCV)
MUT=NXA*NYA
DOXX=XC(NCV) / NXA
ZAZ=ZC(NCV)/NS(NCV)/2.
AC1=FCU2(NCV)*BETA(NCV)/FN2(NCV)/VEL(NCV)
C WRITE(6,2) NSA,NYA,NXA,DOXX,ZAZ,AC1,(SSGAIN(K,NCV),K=1,NAA)
C 2 FORMAT(1H0,3I5,3G12.5/16(1X,8G12.5/))
DO 80 J=1,NYA
IZ=1+(J-1)*NAA
POP = PPD( IZ )/SAT
POP1 = POP * 1.
SGAINX(J) = POP/POP1*DOXX/2.
WINTS(J) = POP/POP1
G(J) = SSGAIN(1,NCV)/POP1*EXP(-AC1*SGAINX(J))
80 GG( IZ ) = EXP(G(J)*ZAZ)
DO 110 I=2,NXA
WRITE(6,3) G(32),SGAINX(32),WINTS(32),GG(I-1,32)
C 3 FORMAT(1X,4G12.5)
DO 110 J=1,NYA
IZ = 1+(J-1)*NXA
POP = PPD( IZ )/SAT
POP1 = 1.*POP
WINT=POP / POP1
SGAINX(J) = SGAINX(J)+(WINT+WINTS(J))/2.*DOXX
WINTS(J) = WINT
G(J) = SSGAIN(1,NCV) /POP1*EXP(-AC1*SGAINX(J))
110 GG( IZ ) = EXP(G(J)*ZAZ)
IF(IUS(NCV).LE. 0) GO TO 300
C
C COMPUTE HEAT RELEASE FUNCTION FOR FUMS ANALYSIS
C
ETA = .40
HCONST=2.E+7*(1.-ETA)/ETA/(ZC(NCV)/NSA)
DO 200 I=1,MUT
BIGG=GG( I )**2
200 HPD( I )=HCONST*HPD( I )*(BIGG-1.0)/(BIGG+1.0)
C 300 CALL CPUFIN(IFIN)
C DELT=(ISMT-IFIN)/100.
C WRITE(6,310) DELT
C 310 FORMAT(25H0 GAIN CALCULATIONS COST 6G12.5,20H SECONDS OF CPU TIME/
C //)
300 RETURN
END
```

The following is from Reference 1 and is included for the convenience of the reader.

The gain coefficient for a gas dynamic laser is described with the aid of a simple three-level model representing a flowing N_2 - CO_2 system interacting with a 10.6μ beam. The relevant energy-level structure is illustrated schematically in Figure 62. The upper (001) and lower (100) laser levels of CO_2 are designated a and b, respectively. The symbols n_a and n_b denote the population densities occupying these levels. The first excited vibrational level of N_2 is nearly resonant with the upper laser level. The population density N is nearly resonant with the upper laser level. The population density N in this level preferentially pumps the upper laser level. Since the ground state CO_2 and N_2 populations, labelled n_o and N_o , are generally large compared to n_a , n_b , and N , the magnitudes of n_o and N_o are relatively unaffected by transitions to and from the excited levels. Accordingly, n_o and N_o may be viewed as constants, i.e., $n_o/N_o = CO_2/x_{N_2} = \text{constant}$ where x_{CO_2} and x_{N_2} are the mole fractions of CO_2 and N_2 .

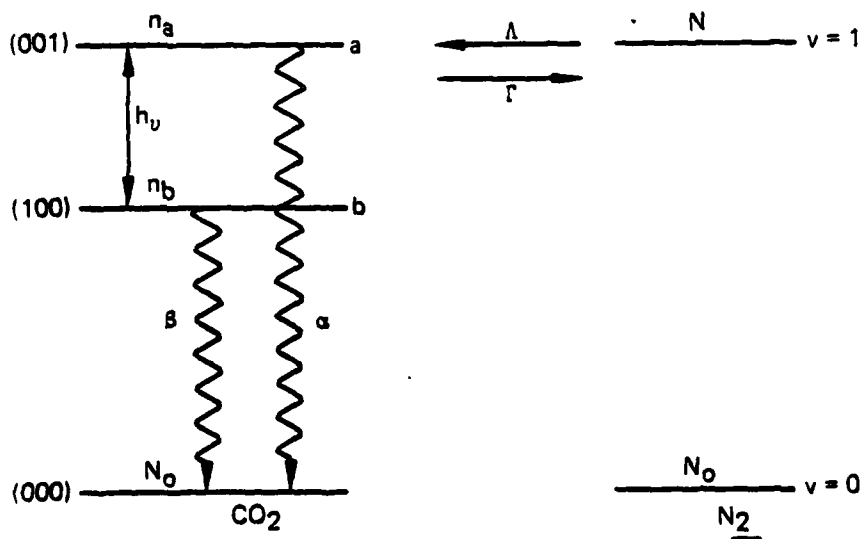


Figure 62. Relevant energy level diagram for N_2 - CO_2 system.

For steady flow in the x-direction the rate equations describing the spatial variation of the three relevant population densities n_a , n_b and N are given by

$$v \frac{\delta n_a}{\delta x} = \Lambda N - (\alpha + \Gamma) n_a - (\sigma I / h\nu) (n_a - n_b) \quad (167)$$

$$v \frac{\delta n_b}{\delta x} = -\beta n_b = (\sigma I / h\nu) (n_a - n_b) \quad (168)$$

$$v \frac{\delta N}{\delta x} = \Gamma n_a - \Lambda N \quad (169)$$

Here, v is the flow velocity (assumed constant); α and β are the relaxation rates of the upper and lower levels; Λ and Γ are the forward and backward pumping rates of the upper laser level; σ is the optical cross section for the laser transition; $h\nu$ is the photon energy; and I is the beam intensity.

Since the pumping rates Λ and Γ are proportional to the ground state population densities n_0 and N_0 , respectively, it follows that

$$\Lambda / \Gamma = x_{\text{CO}_2} / x_{\text{N}_2} \quad (170)$$

Under typical GDL operating conditions $x_{\text{CO}_2} \ll x_{\text{N}_2}$. Also typically, the upper level decay rate is slow relative to the lower level decay rate, and the latter is slow relative to the backward pumping rate, i.e.,

$$\alpha \ll \beta, \Lambda \ll \Gamma \quad (171)$$

The beam is assumed to propagate in the z-direction. For purposes of analysis it is convenient to suppose that the transverse intensity profile at some axial station z can be divided into a series of constant intensity segments, as illustrated in Figure 63. For example, in the n^{th} segment ($x_n < x < x_{n+1}$) the intensity distribution is approximated by the value $I_n =$ constant. For the moment, the segment width $x_{n+1} - x_n$ is left unspecified.

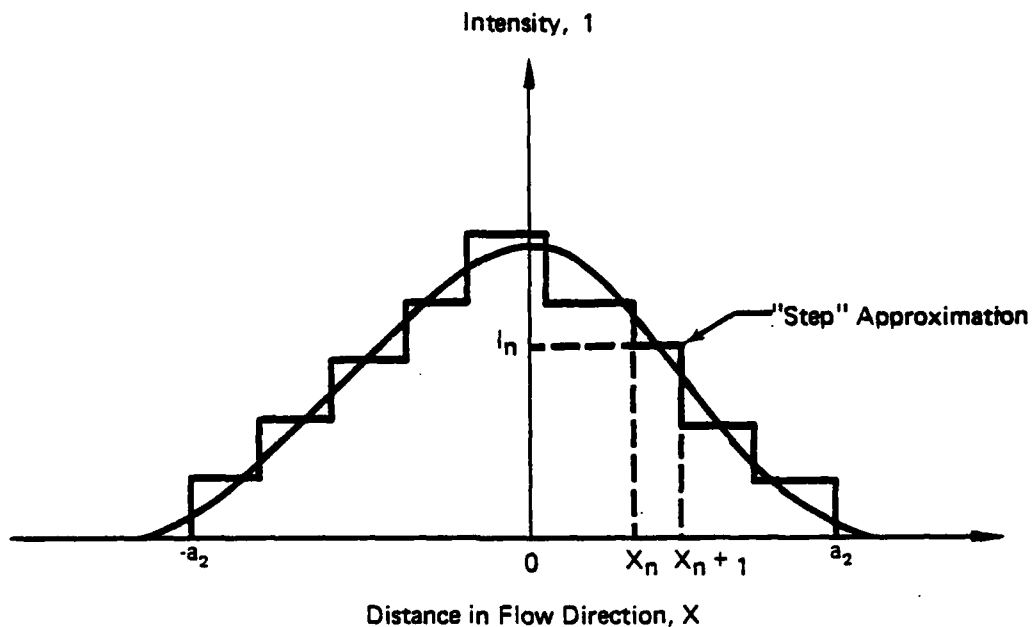


Figure 63. Step approximation to transverse intensity profile.

The gain coefficient for the laser transition is defined by

$$g(x, I) = \sigma(n_a - n_b) \quad (172)$$

We wish to solve for $g = g(x, I)$ in the n^{th} segment ($n = 1, 2, 3, \dots$) where $I = I_n = \text{constant}$. The upstream edge conditions $n_a(x_n)$, $n_b(x_n)$ and $N(x_n)$ are presumed known from the solution in the adjacent upstream segment. By successive application of the n^{th} segment solution, commencing with the segment at the upstream edge of the beam, one can in principle solve for g throughout the optical cavity.

The advantage of the segmented description is that an exact solution can be found in a region of constant beam intensity. Moreover, under suitable approximations, to be discussed later, this sequence of exact solutions can be put in a simple analytical form suitable for application to a smoothly varying beam profile.

Applying the Laplace transform to equations (167) through (169), one obtains

$$\underline{\underline{a}} \underline{\underline{b}} = \underline{\underline{c}} \quad (173)$$

where

$$\underline{\underline{a}} = \begin{Bmatrix} s+\alpha+\Gamma+W_n & -W & -\Lambda \\ -W_n & s+\beta+W_n & 0 \\ -\Gamma & 0 & s+\Lambda \end{Bmatrix}$$

$$\underline{\underline{b}} = \begin{Bmatrix} \tilde{n}_a \\ \tilde{n}_b \\ \tilde{N} \end{Bmatrix} \quad \underline{\underline{c}} = \begin{Bmatrix} n_a(x_n) \\ n_b(x_n) \\ N(x_n) \end{Bmatrix}$$

Here, $\tilde{n}_a(s) = (1/v) \int_{x_n} dx n_a(x) \exp[-s(x - x_n)/v]$, etc..
and $W_n = \sigma I_n/hv$.

Solving by $\underline{\underline{b}}$

$$n_a(s) = |\det|^{-1} [(s+\delta+W_n)(s+\Lambda)n_a(x_n) + W_n(s+\Lambda)n_b(x_n) + \Lambda(s+\beta+W_n)N(x_n)] \quad (174)$$

$$n_b(s) = |\det|^{-1} \left\{ W_s(x+\Lambda)n_a(x_n) + [(s+\alpha+\Gamma+W_n)(s+\Lambda) - \Lambda\Gamma]n_b(x_n) + W_n\Lambda N(x_n) \right\} \quad (175)$$

$$N(s) = |\det|^{-1} \left\{ (s+\beta+W_n)\Gamma n_a(x_n) + W_n\Gamma n_b(x_n) + [(s+\alpha+\Gamma+W_n)(s+\beta+W_n) - W_n^2]N(x_n) \right\} \quad (176)$$

Here, $|\det|$ is the determinant of $\underline{\underline{a}}$ given by

$$|\det| = s^3 + k_2 s^2 + k_1 s + k_0 \quad (177)$$

where

$$k_2 = \beta + \Lambda + \Gamma + 2W_n$$

$$k_1 = \beta(\Lambda + \Gamma) + W_n(2\Lambda + \Gamma + \beta)$$

$$k_0 = \Lambda\beta(\alpha + W_n)$$

The approximate equality sign refers to the use of the first half ($\alpha \ll \beta, \Lambda, \Gamma$) of the inequality 171.

Under the same approximation the roots of equation (177) are given by

$$r_1 \approx \frac{\Lambda \beta (\alpha + W_n)}{\beta (\Lambda + \Gamma) + W_n (2\Lambda + \Gamma + \beta)} \quad (178)$$

$$r_2 = \frac{1}{2} \left[\Lambda + \Gamma + \beta + 2W_n - \sqrt{(\Lambda + \Gamma - \beta)^2 + 4W_n (W_n - \Lambda)} \right] \quad (179)$$

$$r_3 = \frac{1}{2} \left[\Lambda + \Gamma + \beta + 2W_n + \sqrt{(\Lambda + \Gamma - \beta)^2 + 4W_n (W_n - \Lambda)} \right] \quad (180)$$

where $|\det| = (s+r_1) (s+r_2) (s+r_3)$.

In the absence of a beam ($W_n = 0$) the roots r_1 , r_2 and r_3 have a simple physical interpretation.

$$\begin{aligned} r_1 &\rightarrow r_1^0 = \alpha \Lambda / (\Lambda + \Gamma) \\ r_2 &\rightarrow r_2^0 = \beta \\ r_3 &\rightarrow r_3^0 = \Lambda + \Gamma \end{aligned} \quad (181)$$

The value r_1^0 defines the relaxation rate of the available laser energy (the upper laser level coupled to the vibrationally excited N_2) in the absence of a beam; r_2^0 describes the lower level decay; and r_3^0 is the rate at which pumping equilibrium between the excited CO_2 and N_2 is established. Typically, $r_1^0 \ll r_2^0 \ll r_3^0$.

As W_n is increased from zero, the physical identification of the roots r_1 , r_2 , and r_3 becomes somewhat obscure. However, the inequality $r_1 \ll r_2 \ll r_3$ appears to hold for all values of W_n . This feature leads to an important simplification.

*Care must be exercised not to introduce the second inequality at too early a stage in the calculation.

Taking the inverse Laplace transform of equations (174) through (176) one obtains a solution in the form

$$n_a(x) = A \exp[-r_1(x-x_n)/v] + B \exp[-r_2(x-x_n)/v] + C \exp[-r_3(x-x_n)/v] \quad (182)$$

where A, B, and C are functions of the initial conditions $n_a(x_n)$, etc., and of the various rate constants. Similar expressions hold for $n_b(x)$ and $N(x)$.

In the absence of a beam ($W_n = 0$) this solution reduces to the simple form

$$\begin{aligned} n_a(x) = & \frac{\lambda}{\lambda + \Gamma} [n_a(x_n) + N(x_n)] \exp[-r_1^0(x-x_n)/v] \\ & + \left[\frac{\Gamma n_a(x_n) - \lambda N(x_n)}{\lambda + \Gamma} \right] \exp[-r_3^0(x-x_n)/v] \end{aligned} \quad (183)$$

$$n_b(x) = n_b(x_n) \exp[-r_2^0(x-x_n)/v] \quad (184)$$

$$\begin{aligned} N(x) = & \frac{\Gamma}{\lambda + \Gamma} [n_a(x_n) + N(x_n)] \exp[-r_1^0(x-x_n)/v] \\ & - \left[\frac{\Gamma n_a(x_n) - \lambda N(x_n)}{\lambda + \Gamma} \right] \exp[-r_3^0(x-x_n)/v] \end{aligned} \quad (185)$$

The quantity $[n_a(x) + N(x)]$, describing the available laser energy, decays at the characteristic rate r_1^0 while the quantity $[\Gamma n_a(x) - \lambda N(x)]$, describing the departure from pumping equilibrium, decays at the rate r_3^0 .

When the beam intensity I_n is nonvanishing, the details of the solution become rather cumbersome, and successive application of this solution to a series of adjacent beam segments would be a tedious task. Fortunately this complexity can be largely eliminated with the aid of two physically reasonable assumptions.

The first assumption is that the segment widths $\Delta x_n = x_{n+1} - x_n$ can be made somewhat larger than the characteristic lengths v/r_2 and v/r_3 . In other words, the intensity distribution $I = I(x)$ is assumed to vary little over the characteristic lengths for lower level decay and pumping equilibrium. In this event the second and third terms in equation (182), evaluated at the downstream edge of the n^{th} segment, can be neglected.

If, in addition, the rate of stimulated emission W_n ($n = 1, 2, 3, \dots$) is less than the pumping equilibrium rate $\Lambda + \Gamma$, it follows that pumping equilibrium can be assumed throughout the optical cavity, i.e.,

$$\Gamma n_a(x) = N(x) \quad (186)$$

Application of these approximations yields for the population difference between laser levels evaluated at the downstream edge of the n^{th} segment

$$\begin{aligned} n_a(x_{n+1}) - n_b(x_{n+1}) &= \frac{\beta(\Lambda + \Gamma)n_a(x_n)}{\beta(\Lambda + \Gamma) + W_n(2\Lambda + \Gamma + \beta)} \exp\left[-r_1(W_n) \frac{\Delta x_n}{v}\right] \\ &\approx \frac{\beta}{\beta + W_n} n_a(x_n) \exp\left[-r_1(W_n) \frac{\Delta x_n}{v}\right] \end{aligned} \quad (187)$$

where, in the latter expression, use has been made of the second half of the inequality (171).

By a similar procedure one finds

$$n_a(x_n) = n_a(x_{n-1}) \exp\left[-r_1(W_{n-1}) \frac{\Delta x_{n-1}}{v}\right] \quad (188)$$

Repeated substitution of equation (188) into (187) gives

$$\begin{aligned} n_a(x_{n+1}) - n_b(x_{n+1}) &= \frac{\beta n_a(x_1)}{\beta + W_n} \exp\left\{-\left[r_1(W_n) \Delta x_n + r_1(W_{n-1}) \Delta x_{n-1} \right. \right. \\ &\quad \left. \left. + \dots + r_1(W_0) \Delta x_0\right] / v\right\} \end{aligned} \quad (189)$$

If the segment widths Δx_n ($n = 0, 1, 2, \dots$) are now viewed as "infinitesimals" equation (189) may be rewritten

$$n_a(x) - n_b(x) = \frac{n_a(x_0)}{1+w(x)} \exp \left[-\frac{1}{v} \int_{x_0}^x dx' r_1 \right] \quad (190)$$

$$= \frac{n_a(x_0) \exp \left[-r_1^* (x-x_0)/v \right]}{1+w(x)} \exp \left[-\frac{1}{v} \int_{x_0}^x dx' (r_1 - r_1^*) \right]$$

where $w(x) = \sigma I(x)/h\nu\beta$ and x_0 defines a convenient reference station (e.g., the upstream edge of the beam).

Using the basic definition (172), the rate expressions (178) and (181), the identity (170), and the inequality (171), one finds on substitution into (190)

$$g(x) = \left[\frac{g_0(x)}{1+w(x)} \right] \exp \left\{ -\frac{x_{CO_2} \beta}{x_{N_2} v} \int_{x_0}^x dx' \frac{w(x')}{1+w(x')} \right\} \quad (191)$$

where g_0 is the small-signal gain coefficient given by

$$g_0(x) = g_0(x_0) \exp \left[-\frac{x_{CO_2} \alpha (x-x_0)}{x_{N_2} v} \right] \quad (192)$$

It is instructive to note the physical significance of various terms appearing in equations (191) and (192). The term in square brackets in equation (191) is analogous to the usual gain expression for a homogeneously broadened line in a nonflowing laser medium. Here, however, the small-signal gain coefficient (192) is not constant, but decays exponentially with distance downstream. The nondimensional intensity $w(x)$ measures the rate of simulated emission $\sigma I/h\nu$ relative to the decay rate β of the lower level. For a nonflowing laser the value $w = 1$ defines the saturation intensity of the medium.

The exponential factor in equation (191) represents a corrective term due to flow. The probability that an initially excited CO_2 molecule will remain excited after traversing a beam is dependent on the beam profile encountered by the molecule upstream of the point in question. This explains the presence of an integral over the upstream flowpath in equation (191).

In summary, a simple approximate expression has been derived for the gain coefficient in a flowing $\text{N}_2\text{-CO}_2$ system. The validity of this expression rests on two principal assumptions: (1) instantaneous pumping equilibrium is maintained throughout the optical cavity and (2) the beam intensity changes slowly over the characteristic distance for lower level decay. Although these conditions are not always satisfied in practice, particularly near the upstream edge of the beam, it is believed that even in these instances equation (191) provides a qualitatively accurate description of gain saturation in a GDL. The gain coefficient defined by equation (191) is then included in the complex transmission function

$$t = \exp \left[g(x,y;I) \Delta L/2 + i\Delta\phi(x,y;I) \right] \quad (193)$$

to describe the effect of the medium gain throughout a segment of length ΔL . Here, $\Delta\phi$ represents a phase shift due to possible refractive index variations.

30. SUBROUTINE SLIVER

a. Purpose -- Subroutine SLIVER, shown in Figure 64, applies an annular aperture to the field. It can be centered anywhere in the mesh.

b. Relevant formalism -- The field is set to zero interior to the annular aperture. Mesh squares intersecting the aperture edge have the field linearly adjusted for the relative area intersected by the aperture edge.

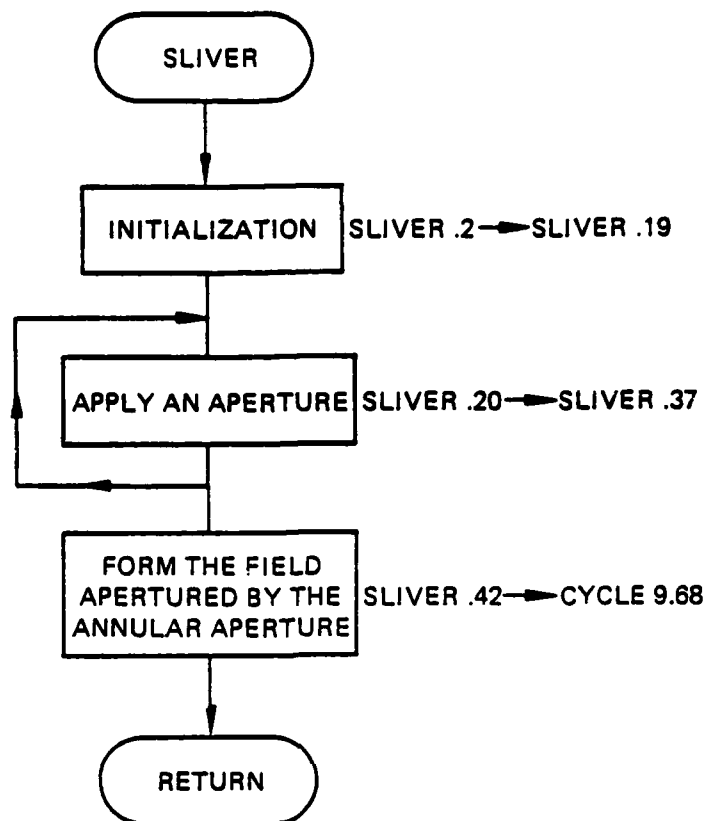


Figure 64. Subroutine SLIVER organization.

c. Fortran

Arguments

RIN = Radius of the OUTER edge of the annulus (cm)

ROUT = Radius of the INNER edge of the annulus (cm)

NOTE: Both RIN and ROUT must be negative to call "SLIVER" since if DOUT (=2*RIN) and DIN (=2*ROUT) are negative in the GDL call IFLOW = 4 section SLIVER is called instead of APRTR.

Common Variables Altered

CFIL = CFIL contains the original field

CU = CU is used to find the aperture field.

The Logic of Subroutine SLIVER is the following:

The final field is formed by subtracting an apertured field from the original. The aperture has a center disk of radius ROUT while the inner radius of the outer edge is RIN.

The center obscuration is first removed (IIN=0), then the outer obscuration (IIN=1). This apertured field (CU) is then subtracted from the original field (stored in CFIL) to form the field apertured by the annular aperture (CU).

The SLIVER subroutine computer printout follows.

SUBROUTINE SLIVER 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE SLIVER(MIN,ROUT,XPOS,YPOS)            SLIVER    2
ANNULAR APERTURE TRANSMISSION FUNCTION        SLIVER    3
THIS ROUTINE, WHICH OPERATES IN A MANNER SIMILAR TO SUBROUTINE        SLIVER    4
APERTURE, APPLIES AN ANNULAR OBSCURATION WITH INNER AND OUTER        SLIVER    5
RAII OF RIN AND ROUT, RESPECTIVELY            SLIVER    6
LEVEL 2. CU                                      SLIVER    7
COMMON/MELT/CU(10384),CFIL(10312),XAH(128),NL,NPTS,NPY,UNX,UNY        SLIVER    8
COMPLEX CU,CFIL                                 SLIVER    9
RD(XX,YY,IX,IY)=SUMT((ABS(XX)*(X*UX/2.)*2+(ABS(YY)*UY/2.*IY)*2)        SLIVER  10
HAPHTH=ABS(ROUT)                               SLIVER  11
NDISK=ABS(RIN)                                 SLIVER  12
OX=XAH(2)-XAH(1)                               SLIVER  13
OY=OAX                                         SLIVER  14
IIN=0                                         SLIVER  15
HAX=HAPHTH                                    SLIVER  16
NOB=NPTS*NPY                                 SLIVER  17
DO 98 I=1,NOB                                 SLIVER  18
98 CFIL(I)=CU(I)                                SLIVER  19
99 DO 101 IIX=1,NPTS                         SLIVER  20
X=XAH(IIX)-UNX-XPOS                         SLIVER  21
DO 101 IY=1,NPY                              SLIVER  22
Y=XAH(IY)-UNY-YPOS                         SLIVER  23
RPP=RD(X,Y,1,1)                              SLIVER  24
RMM=RD(X,Y,-1,-1)                            SLIVER  25
RMP=RD(X,Y,-1,1)                             SLIVER  26
RPM=RD(X,Y,1,-1)                             SLIVER  27
PER=1.                                        SLIVER  28
RMAX=AMAX1(RPP,RMM,RMP,RPM)                 SLIVER  29
IF (RMAX.LE.HAX) GO TO 100                   SLIVER  30
PER=0.                                        SLIVER  31
RMIN=AMIN1(RMP,RMM,RMP,RPM)                 SLIVER  32
IF (RMIN.GE.HAX) GO TO 100                   SLIVER  33
PER=(HAX-RMIN)/(RMAX-RMIN)                  SLIVER  34
100 IF (IIN.EQ.1) PER=1.-PER                SLIVER  35
NNN = IIX+(IY-1)*NPTS                      SLIVER  36
101 CU(NNN) = CU(NNN) * (1.-SUMT(PER))        SLIVER  37
IF (NDISK.EQ.0..OR.IIN.EQ.1) GO TO 102      SLIVER  38
IIN=1                                        SLIVER  39
RAD=NDISK                                    SLIVER  40
GO TO 99                                     SLIVER  41
102 DO 103 I=1,NOB                           SLIVER  42
CU(I) = CFIL(I)-CU(I)                        SLIVER  43
103 CONTINUE                                 SLIVER  44
WRITE(6,300) RAPHTH,NDISK                   SLIVER  45
300 FORMAT (//28H ANNULAR OBSCURATION APPLIED /18H INSIDE RADIUS=,        SLIVER  46
X F10.3,17H OUTSIDE RADIUS=,F10.3)        SLIVER  47
RETURN                                        SLIVER  48
END                                            SLIVER  49

```

31. SUBROUTINE SPIDER

a. Purpose -- The SPIDER subroutine shown in Figure 65 applies an obscuration to the complex amplitude field in the form of several support struts, such as those used in a Cassegrain telescope system. Up to six struts at separate angles may be modeled. The result of the obscuration is listed in the output stream as an aperture loss.

b. Relevant formalism -- An angular deviation limit α calculated from the obscuration inside diameter d , the grid spacing Δx , and the strut width w , according to

$$\alpha = \sin^{-1} (w+2\Delta x)/d \quad (194)$$

Field points whose inclination angle is not within $\pm\alpha$ of a strut angle are assumed to be unobscured. Those points falling within this limit are subjected to closer inspection.

The distance δ from a grid center (x,y) to the strut centerline is calculated by

$$\delta = |y \cos\theta - x \sin\theta| \quad (195)$$

where θ is the strut angle. The half-width of a grid measured along a normal to the strut h is calculated by

$$h = x/2./\text{AMAX} (|\sin\theta|, |\cos\theta|) \quad (196)$$

then the maximum and minimum distance of the grid area from the centerline, d_{\max} and d_{\min} are

$$\begin{aligned} d_{\max} &= \delta + h \\ d_{\min} &= \delta - h \end{aligned}$$

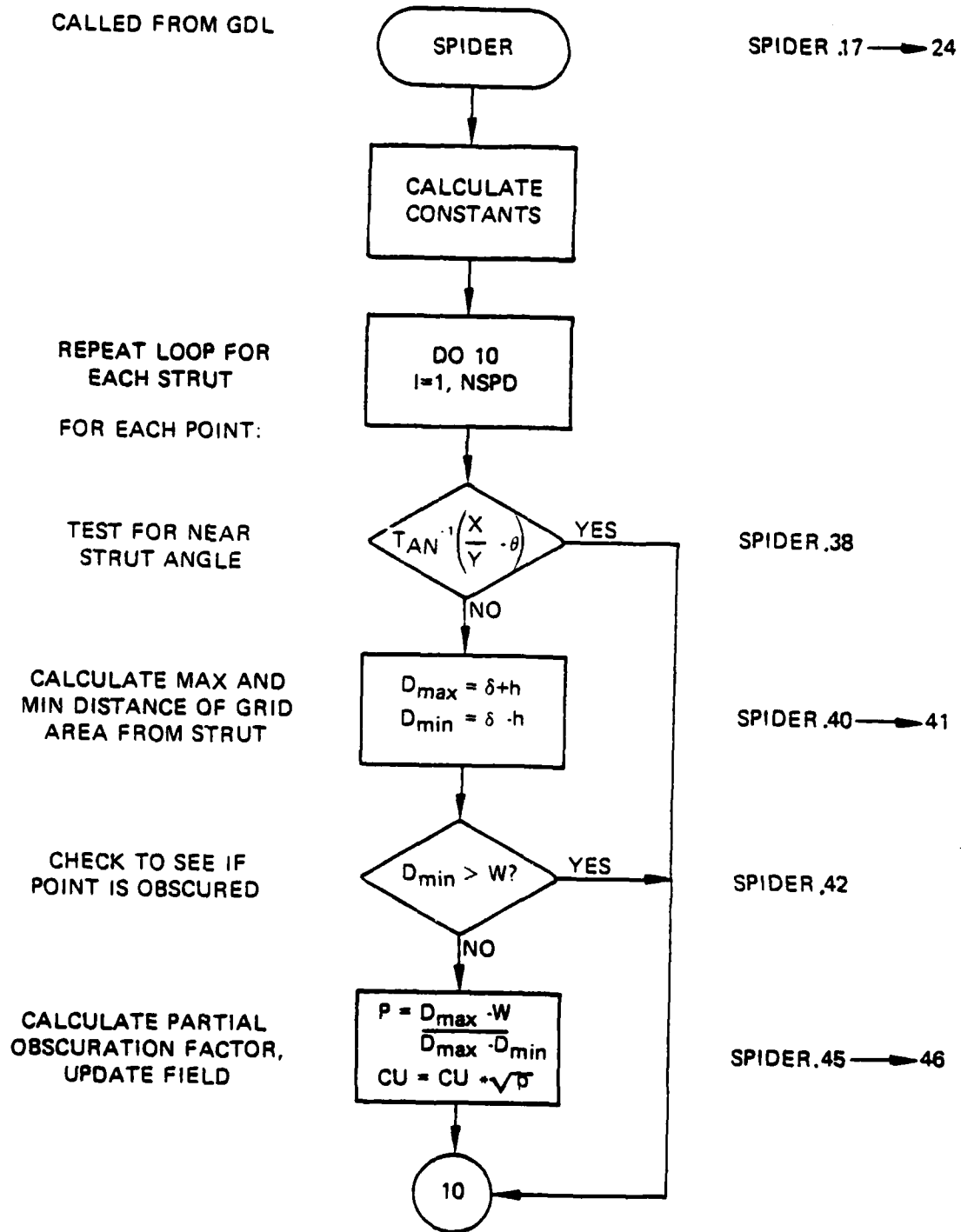


Figure 65. Subroutine SPIDER flow chart.

Points where d_{\min} is greater than the strut half width h_s are not obscured.
 Points where d_{\max} is less than the strut half width are totally obscured.
 The intensity of all other points is weighted according to

$$\text{intensity weighting} = (d_{\max} - h_s) / (d_{\max} - d_{\min}) \quad (197)$$

Argument List

DIH diameter of inner edge of support (hub)
 NSPD number of struts or spokes
 THETA array of strut angles
 WIDTH strut width
 XC x-position of center of obscuration
 YC y-position of center of obscuration

Relevant Variables

ANG inclination angle of a point (x,y)
 ANGTOL angular width about the strut angle which defines the region
 to be searched for possible obscuration
 DELTA distance from (x,y) to the strut along a normal
 DELXDH half-width of coordinate grid measured along a normal to a
 strut
 PER weighting factor in establishing fractional obscuration

Commons Modified

/MELT/
 CU the complex amplitude field.

The SPIDER subroutine computer printout follows.

SUBROUTINE SPIDER 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE SPIDEM (WIDTH,THETA,NSPD,XC,YC,DIH)
C    GENERAL SUPPORT STRUT MODEL
C    ***** MODIFIED 10/17/75 TO HANDLE MULTIPLE THETAS *****
C    THIS ROUTINE APPLIES AN OBSCURING STRUT TRANSMISSION FUNCTION TO
C    THE COMPLEX FIELD. THE STRUT IS WIDTH WIDE WITH AN ANGLE THETA
C    (IN THE BEAM COORDINATE SYSTEM) AND GUES RADially OUTWARD FROM
C    LOCATION (XC,YC). DIH IS HUB DIAMETER,NSPD IS NO. OF STRUTS.
C    DELXDH IS WIDTH/2 OF COORDINATE GRID ALONG NORMAL TO STRUT.
C    DELTA IS DISTANCE FROM X,Y TO CENTER OF STRUT ALONG NORMAL
C    TO STRUT.
C    SPIDEM        2
C    SPIDEM        3
C    SPIDEM        4
C    SPIDEM        5
C    SPIDEM        6
C    SPIDEM        7
C    SPIDEM        8
C    SPIDEM        9
C    SPIDEM       10
C    SPIDEM       11

```

LEVEL 2, CU	SPIDEN	12
COMMON/MELT/CU(1038*),CFIL(10512),X(128),WL,NPTS,NPY,UMX,UMY	SPIDEN	13
DIMENSION THETA(1),THET(6),SINT(6),CUST(6),DELXUM(6)	SPIDEN	14
COMPLEX CU,CFIL	SPIDEN	15
DATA PI,TWOP/ 3.141593,6.283186 /	SPIDEN	16
WOTHM = #DTHM/2.0	SPIDEN	17
DELXU2 = (X(2) - X(1)) / 2.	SPIDEN	18
ANGTUL = ASIN ((WOTHM*2.0*(X(2)-X(1))) / DIM)	SPIDEN	19
DO 5 IT=1,NSPD	SPIDEN	20
THET(IT) = THETA(IT)/57.3	SPIDEN	21
SINT(IT) = SIN(THET(IT))	SPIDEN	22
CUST(IT) = COS(THET(IT))	SPIDEN	23
5 DELXUM(IT) = DELXU2 / AMAX(ABS(CUST(IT)),ABS(SINT(IT)))	SPIDEN	24
IZ=0	SPIDEN	25
DO 10 J=1,NPY	SPIDEN	26
DO 10 I=1,NPTS	SPIDEN	27
IZ = IZ+1	SPIDEN	28
ANG = ATAN2(X(J),X(I))	SPIDEN	29
C THIS STATEMENT CHANGES THE ATAN2 RETURNED ANGLE FROM THE INTERVAL	SPIDEN	30
C -PI TO +PI TO THE INTERVAL 0 TO 2PI.	SPIDEN	31
IF (ANG.GT.(-PI).AND.ANG.LT. 0.) ANG = ANG + TWOP	SPIDEN	32
DO 10 IT=1,NSPD	SPIDEN	33
C THE FOLLOWING IS NECESSARY TO MAKE ANGLES NEAR 2PI SEEM CLOSE TO	SPIDEN	34
C ANGLES NEAR 0.	SPIDEN	35
IF (ANG.LT. -PI) GO TO 15	SPIDEN	36
IF (ABS(ANG-TWOP)-THET(IT)).LE.ANGTUL) GO TO 17	SPIDEN	37
15 IF(ABS(ANG-THET(IT)).GT.ANGTUL) GO TO 10	SPIDEN	38
17 DELTA = ABS((X(J)-YC)*COST(IT)-(X(I)-XC)*SINT(IT))	SPIDEN	39
DMAX = DELTA*DELXUM(IT)	SPIDEN	40
DMIN = DELTA-DELXUM(IT)	SPIDEN	41
IF(DMIN.GE.#DTHM) GO TO 10	SPIDEN	42
PEH = 0.0	SPIDEN	43
IF(DMAX.LE.#DTHM) GO TO 20	SPIDEN	44
PEH = SQRT((DMAX-#DTHM)/(DMAX-DMIN))	SPIDEN	45
20 CU(IZ) = CU(IZ)*PEH	SPIDEN	46
10 CONTINUE	SPIDEN	47
RETURN	SPIDEN	48
END	SPIDEN	49

32 SUBROUTINE SPTAN

The SPTAN subroutine shown in Figure 66 functions to take input values of x and y and return the angle whose tangent they represent. SPTAN insures that the angle returned is within the range

$$0 \leq \theta \leq 2\pi$$

FUNCTION SPTAN 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

FUNCTION SPTAN(X,Y)	SPTAN	2
PI=3.14159265	SPTAN	3
SPTAN=0.0	SPTAN	4
IF(X) 10,20,30	SPTAN	5
10 SPTAN=PI*ATAN(Y/X)	SPTAN	6
RETURN	SPTAN	7
20 IF(Y) 21,22,23	SPTAN	8
21 SPTAN=1.5*PI	SPTAN	9
22 RETURN	SPTAN	10
23 SPTAN=0.5*PI	SPTAN	11
RETURN	SPTAN	12
30 SPTAN=ATAN(Y/X)	SPTAN	13
IF(Y.LT.0.0) SPTAN=SPTAN+2.0*PI	SPTAN	14
RETURN	SPTAN	15
END	SPTAN	16

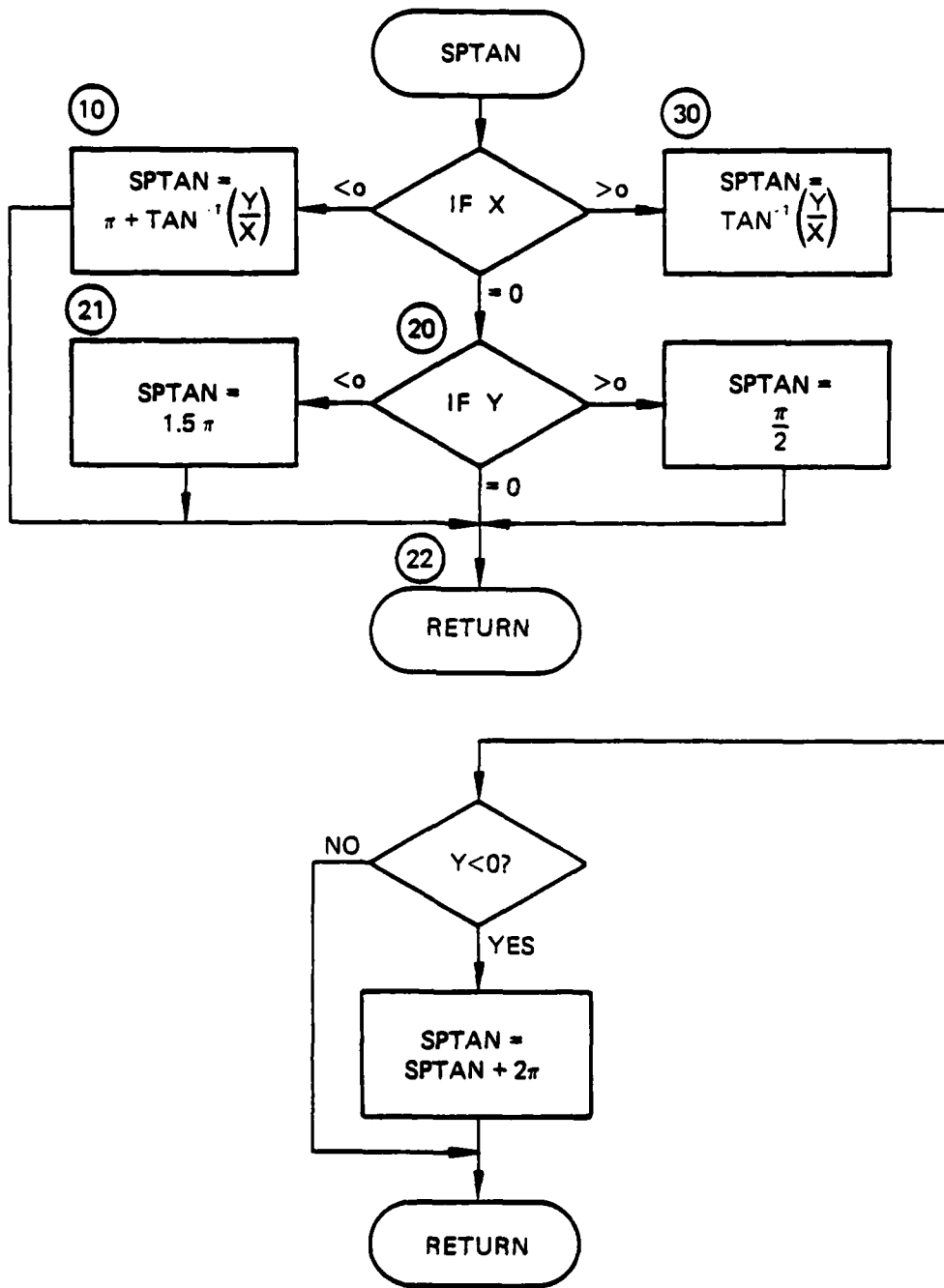


Figure 66. Subroutine SPTAN flow chart.

33. SUBROUTINE STEP

a. Purpose -- Subroutine STEP shown in Figure 67 is used to propagate the field through a vacuum. It also calculates Strehl intensity.

b. Relevant formalism

(1) Propagation -- STEP allows for two types of propagation

(a) Constant area mesh -- This type is used to propagate collimated and quasi-collimated beams. It assumes that edge spreading of the beam due to diffraction is not severe enough for the beam to get too close to the edge of the calculation region.

(b) Variable area mesh (VAMP) -- VAMP is used to propagate beams containing phase with curvature. As will be shown, the curvature is first removed from the field. The (collimated) field is then propagated an equivalent propagation distance which is defined by the formalism. After propagation, the propagated curvature is returned to the field.

The theory of VAMP propagation is developed in Section 5-D of AWFL-TR-73-231 and is repeated here for continuity.

First, consider constant area mesh propagation. The scalar wave function propagating in the Z-direction is written

$$\psi(\vec{x}, t) = U(\vec{x}) e^{i(\omega t - kz)} \quad (198)$$

The function $\psi(x, t)$ obeys the scalar wave equation derived from Maxwell's equations

$$\nabla^2 \psi = \frac{1}{c^2} \frac{\partial^2 \psi}{\partial t^2} \quad (199)$$

If one assumes that

$$\frac{\partial^2 \psi}{\partial t^2} \ll k \frac{\partial u}{\partial z} \quad (200)$$

then $u(x)$ obeys the paraxial wave equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - 2ik \frac{\partial u}{\partial z} = 0 \quad (201)$$

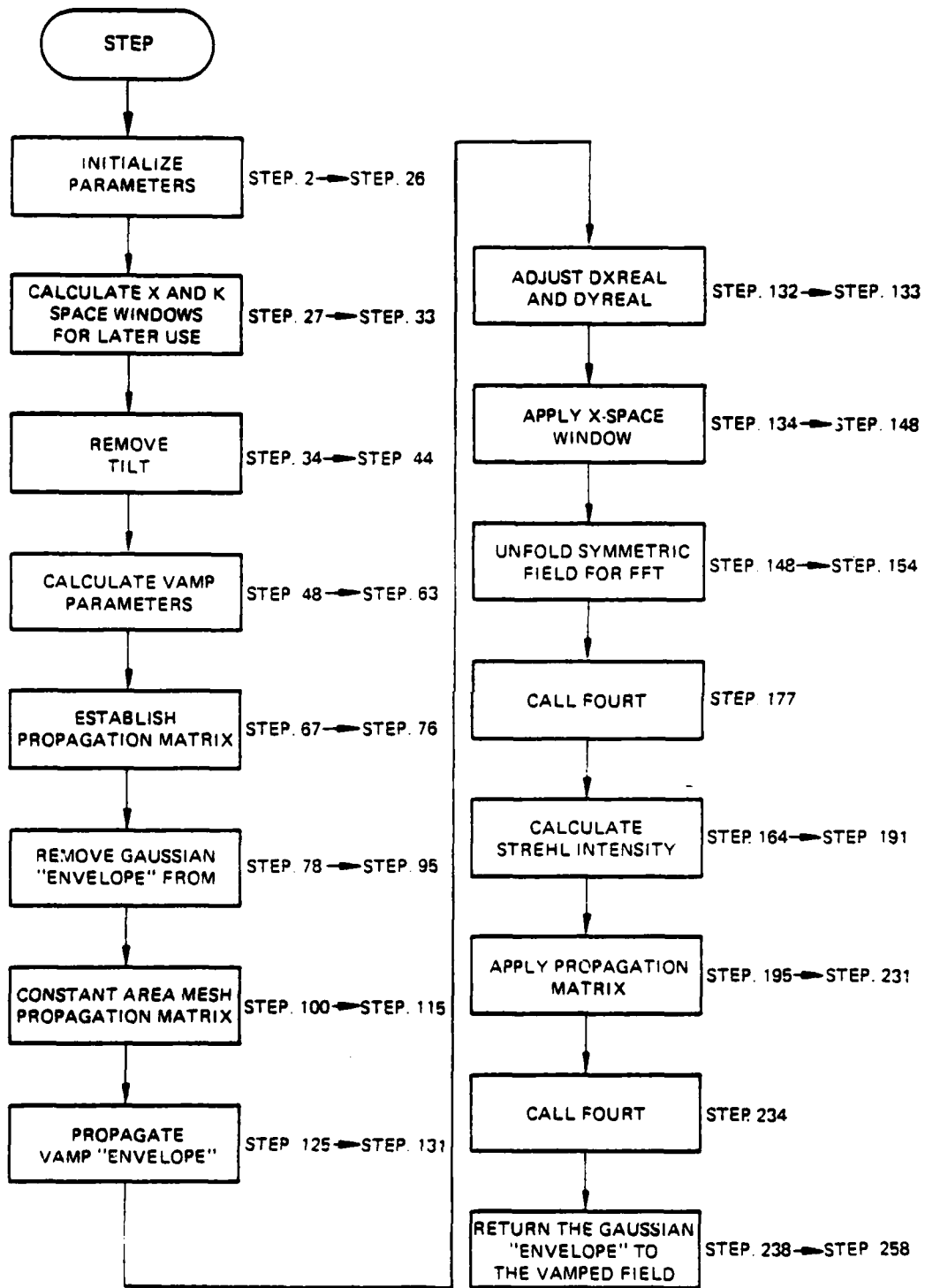


Figure 67. Subroutine STEP organization.

By using the method of Fourier Transforms $u(x)$ is

$$u(\vec{x}) = \iint_{-\infty}^{\infty} df_x df_y e^{2\pi i(f_x x + f_y y)} U(f_x, f_y) e^{i\pi \lambda Z (f_x^2 + f_y^2)} \quad (202)$$

where

$$U(f_x, f_y) = \iint_{-\infty}^{\infty} dx' dy' e^{-2\pi i(f_x x' + f_y y')} U(x', y', 0)$$

The Fourier Transforms are efficiently performed by using the FFT.

For variable area mesh, the following approach is used:

The spreading of the beam is estimated by that of a Gaussian reference beam with the same radius of curvature as the physical beam. This curvature is removed so that during propagation the beam continues to fill the calculation region.

Propagation of a Gaussian beam is easily handled by assuming knowledge of the associated Gaussian plane wave. According to Siegman, Chapter 8, (Ref. 14), a Gaussian plane wave (at $Z = 0$)

$$U_0(x_0, y_0) = \sqrt{\frac{2}{\pi}} \left(\frac{1}{w_0} \right) e^{-(x_0^2 + y_0^2)/w_0^2} \quad (205)$$

when propagated a distance Z becomes

$$u(x, y, z) = \sqrt{\frac{2}{\pi}} \left(\frac{1}{w(z)} \right) e^{-i(kz - \psi(z))} e^{-(x^2 + y^2)} \left(\frac{k}{2R(z)} + \frac{1}{w(z)^2} \right) \quad (204)$$

where

$$R(z) = z + \frac{z_R^2}{z} \quad \psi(z) = \tan^{-1} \left(\frac{z}{z_R} \right)$$

$$w(z) = w_0 \sqrt{1 + \left(\frac{z}{z_R} \right)^2}$$

14. Siegman, A. E., An Introduction to Lasers and Masers, McGraw-Hill, New York, 1971.

with

$$z_R = \frac{\pi w_0^2}{\lambda}, \text{ the Rayleigh range.}$$

Therefore, to propagate a Gaussian beam of waist $w(Z)$ and radius or curvature $R(Z)$ a distance ΔZ , the following approach should be taken:

Knowing the waist and radius of curvature, one can determine the spot size w_0 and distance to the spot size Z , according to

$$z_1 = \frac{R(z_1)}{1 + \left(\frac{\lambda R(z_1)}{\pi w(z_1)} \right)^2} \quad (205)$$

$$w_0 = \frac{w(z_1)}{\sqrt{1 + \left(\frac{\pi w(z_1)}{\lambda R(z_1)} \right)^2}} \quad (206)$$

Then, from this origin a distance $Z_2 = Z_1 + \Delta Z$ is propagated to determine the desired wave function.

Since it is known how a Gaussian wave propagates, it is possible that transforming a given wave with a spherical wave front to Gaussian coordinates could result in the propagation of a quasi-collimated wave. The appropriate transformation is found to be

$$U(\vec{x}) = \frac{V(\vec{x})}{w(z)} e^{i \left[\frac{k(x^2 + y^2)}{2R(z)} + \tan^{-1} \left(\frac{z}{z_R} \right) \right]} \quad (207)$$

where Z is the distance from the current reference Gaussian beam, defined by $R(Z)$ and $w(Z)$ to its spot. z_R is the Rayleigh range of this reference Gaussian beam.

By transforming to Gaussian coordinates:

$$X = x/w(z) \quad Z = \tan^{-1} \left(\frac{z}{z_R} \right) \quad Y = y/w(z) \quad (208)$$

The beam transformation is written as

$$u(\vec{x}) = v(\vec{x}) \frac{\cos z}{w_0} e^{-i(X^2+Y^2) \tan z + iz} \quad (209)$$

Inserting this equation into the paraxial wave equation results in the following differential equation in terms of Gaussian coordinates

$$-4i \frac{\partial v}{\partial z} + \frac{\partial^2 v}{\partial X^2} + \frac{\partial^2 v}{\partial Y^2} + 4(1-(X^2+Y^2))v = 0 \quad (210)$$

which, except for the quadratic, is similar to the paraxial wave equation. The quadratic term $(X^2 + Y^2)v$ can be dropped if the reference Gaussian parameters and propagation distance are chosen so that v is equal to zero whenever X or Y approaches 1. This implies that the initial waist of the reference Gaussian be much larger than the size of the beam to be propagated. The propagation distance ΔZ must then be restricted so that the waist of the reference beam remains large compared with the beam size throughout the propagation. With these restrictions, the equation for v in Gaussian coordinates becomes

$$\frac{\partial^2 v}{\partial X^2} + \frac{\partial^2 v}{\partial Y^2} + 4v - 4i \frac{\partial v}{\partial z} = 0 \quad (211)$$

As is the collimated case, Fourier Transform analysis gives the following result:

$$v(X, Y, Z) = \iint_{-\infty}^{\infty} df_x df_y V(f_x, f_y, Z) e^{2\pi i(f_x X - f_y Y)} \quad (212)$$

where

$$V(f_x, f_y, Z) = V(f_x, f_y, Z_1) e^{-i \left[1 - \pi^2 (f_x^2 + f_y^2) \right] (Z - Z_1)}$$

and

$$V(f_x, f_y, z_1) = \iint_{-\infty}^{\infty} dx dy V(X, Y, Z) e^{-2\pi i (f_x X + f_y Y)}$$

the propagated wavefunction is then $v(X, Y, Z)$ multiplied by the propagation envelope:

$$u(x, y, z) = V(X, Y, Z) \frac{\cos Z}{w_0} e^{i(X^2 + Y^2) \tan Z + iZ} \quad (213)$$

where

$$X = \frac{x}{w(z)} \quad Y = \frac{y}{w(z)} \quad z_1 = \tan^{-1} \left(\frac{z}{z_R} \right)$$

z being the final distance from the reference spot. If the propagation takes place well outside of the Rayleigh range, Z is much greater than Z_R and the expansion of the arctangent for large argument can be used:

$$\begin{aligned} z - z_1 &= \tan^{-1} \left(\frac{z}{z_R} \right) - \tan^{-1} \left(\frac{z_1}{z_R} \right) \\ &= \left(\frac{\pi}{2} - \frac{z_R}{z} \right) - \left(\frac{\pi}{2} - \frac{z_R}{z_1} \right) \\ &= z_R \left(\frac{1}{z_1} - \frac{1}{z} \right) \end{aligned} \quad (214)$$

(2) Strehl intensity -- Since subroutine STEP propagates the beam using Fourier Transforms, the Strehl intensity is easily calculated.

The Strehl intensity gives an irradiation of the amount of aberration present in the beam at a given limiting aperture. It is defined as follows: Consider a field $U(x, y)$. The field in the Fraunhofer diffraction region (the far field) is given by equations (4) through (13) in Goodman:

$$\vec{u}(x) = e^{ikz} \frac{i k}{2z} (x^2 + y^2) \iint_{-\infty}^{\infty} u(\vec{x}') e^{\frac{2\pi i}{\lambda z} (\vec{x} \cdot \vec{x}')} d\vec{x}' \quad (215)$$

Aside from the phase factor in front, this is just the Fourier Transform of the apertured field evaluated at

$$\vec{f} = \frac{\vec{x}}{\lambda z} \quad (216)$$

The Strehl intensity is defined as the ratio of the centerline intensity of the far field to that of a plane wave propagated the same distance coming from the same aperture with the same power. Analytically this is given as

$$I_{\text{STREHL}} = \frac{I_{\text{CL-FF}}}{I_{\text{LL-PW-FF}}} = \frac{\left| F(u(\vec{x}')) \right|_{\vec{f}=0}^2}{\left| F(u_{\text{pw}}(\vec{x}')) \right|_{\vec{f}=0}^2} \quad (217)$$

The plane wave centerline intensity is evaluated from

$$\begin{aligned} F(u_{\text{pw}}(\vec{x}')) &= A_0 \int_0^a r dr \int_0^{2\pi} d\theta e^{2\pi i f_r \cos\theta} \Big]_{\vec{f}=0} \\ &= \pi a^2 A_0 \end{aligned} \quad (218)$$

A_0 being the plane wave amplitude and a the radius of the aperture. Assuming a calculation region size of the $L \times L$ with $N \times N =$ total number of points, the centerline intensity of the far field for the real beam is found from

$$\begin{aligned} F(u(\vec{x}')) &= \iint_{-\infty}^{\infty} dx u(\vec{x}) e^{2\pi i \vec{f} \cdot \vec{x}} \\ &= \int_0^L dx \int_0^L dy u(x) e^{2\pi i \vec{f} \cdot \vec{x}} \\ &\approx \sum_{I=1}^N \left(\frac{L}{N}\right) \sum_{J=1}^N \left(\frac{L}{N}\right) U(I, J) e^{2\pi i \left(\frac{L}{N}\right) (I f_x + J f_y)} \end{aligned} \quad (219)$$

where

$$\Delta x = \Delta y = \frac{L}{N} \quad \text{and} \quad x = I * \left(\frac{L}{N}\right) \quad y = J * \left(\frac{L}{N}\right)$$

assume

$$f_x = \frac{KB}{N} \quad \text{and} \quad f_y = \frac{MB}{N}$$

where B is twice the maximum frequency of the spectrum of u,

then

$$F(u(\vec{x}')) \approx F(K,M) = \left(\frac{L}{N}\right)^2 \sum_{I=1}^N \sum_{J=1}^N U(I,J) \epsilon^{2\pi i \left(\frac{LB}{N}\right) \left(\frac{KI}{N} + \frac{MJ}{N}\right)} \quad (220)$$

But from the theory of discrete Fourier Transforms $LB = N$, so

$$F(K,M) = \left(\frac{L}{N}\right)^2 \sum_{I=1}^N \sum_{J=1}^N U(I,J) \epsilon^{2\pi i (KI + MJ)/N} \quad (221)$$

The whole sum is just the (K,M) output of the FFT routine, so

$$F(K,M) = \left(\frac{L}{N}\right)^2 F_{\text{FFT}}(K, M) \quad (222)$$

The FFT returns the DC value (centerline) at $F_{\text{FFT}}(1,1)$ so the Strehl intensity is defined as

$$I_{\text{STREHL}} = \frac{\left(\frac{L}{N}\right)^4 \left| F_{\text{FFT}}(1,1) \right|^2}{(\pi a^2)^2 I_0} \quad (223)$$

where $I_0 = A_0^2 =$ plane wave intensity.

Note: If the beam is not limited by an exit aperture just before the Strehl calculations, it is possible to have I_{STREHL} greater than one.

c. Fortran

Argument List

- DELZ = Distance to be
- RADCY = radius of curvature or the phase front
- WINDOX = x-space cosine data window for FFT
- WINDOK = K-space cosine data window for FFT
- IFG = Vamp control parameter
- = 1 constant mesh
- = 2 variable mesh
- ITR = Vamp control parameter
- = 0 stay in vamp
- = 1 transform back to constant mesh space
- IPS = Tilt and defocus removal flag
- = 0 no correction
- = 1 remove tilt
- = 2 find defocus radius of curvature
- = 3 1 + 2 together
- AX }
AY } = total beam tilt keep track of for beam placement in the
inertial coordinate system instead of the beam coordinate
system
- NWRT ≠ 0 Propagates a wave distance DELZ without altering the stored
value of total Z. NWRT = 1. Suppresses Strehl intensity cal-
culation as well. NWRT = 1 when STEP is called from QUAL.
- IFLAG ≠ 0 Assumes VAMP and/or CAMP parameters are established. It
tells the routine to continue the propagation based on
previous calculations of waist and curvature.

Common Variables Altered:

- CU - becomes the propagated field
- CFIL - is altered if IPS ≠ 0 by a call to TILT

- X - altered if in VAMP
- DXREAL - moved to keep track of center of beam in inertial frame as
DYREAL the beam propagates
- WNOW - VAMP parameter altered to keep track of the current spot size
- NREG - Flag to tell whether:
 - = 0: Constant area mesh propagation
 - = 1: VAMP inside half the Rayleigh range
 - = 2: VAMP outside twice the Rayleigh range

Other routines called:

TILT

FOURT

Computer printouts for subroutine STEP follow.

SUBROUTINE STEP 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE STEP(DELZ,HADC,H,WNDOX,WNUOX,IFG,ITR,IPS,AX,AY,NWHT, STEP      2
X IFLAG) STEP      3
C GENERAL PROPAGATING ALGORITHM STEP      4
C THIS ROUTINE IS USED TO PROPAGATE THE COMPLEX FIELD A DISTANCE STEP      5
C DELZ = IFLAG=1 IS USED WHEN CONTINUING WITH SAME PROPAGATING MATRIX STEP      6
LEVEL 2: CU,CUM STEP      7
COMMON/WAY/WNOW,NREG,HAPTH STEP      8
COMMON/MELT/CU(16384),CFIL(16512),X(128),WL,NPTS,NPY,DXREAL,DYREAL STEP      9
DIMENSION NNO(2),APH(2,2610),PACTH(64),CUM(32768),CUUMH(2) STEP     10
DOUBLE PRECISION WU,ZHAL,ZI,HADCUN,WU,WFTNZZ STEP     11
COMPLEX CU,CFIL,CUUM STEP     12
EQUIVALENCE (CU(1),CUM(1)), (CUUM,CUUMH(1)) STEP     13
DATA ZINTE /0.0/ STEP     14
IF (IFLAG.NE.0) GO TO 2000 STEP     15
PI=3.141592 STEP     16
NP2P2=NPTS*2 STEP     17
NP = NPTS/2 STEP     18
NPP1= NP *1 STEP     19
ANP2=1.0/FLOAT(NPTS)**2 STEP     20
NNU(1) = NPTS STEP     21
NNU(2) = NPTS STEP     22
NAN=2*NPTS*NPTS STEP     23
NREG=0 STEP     24
HADCUN=HADC,H STEP     25
DCALC1=X(NPTS)-X(1)+X(2)-X(1) STEP     26
IF (WNDOX.LE.0.0) GO TO 48 STEP     27
WNUOX = WNDOX*FLOAT(NPTS) STEP     28

```

C	X=SPACE COSINE DATA WINDOW	STEP	29
	UU 211 I=1,NWNUOX	STEP	30
211	FACTH(I) = (1.0-COS(PI*FLOAT(I)/FLOAT(NWNUOX)))/2.0	STEP	31
*8	NWNUOK = NWNUOK*FLOAT(NPTS)	STEP	32
	N0=NPP1-1-NWNUOK	STEP	33
	IF (IPS.NE.0) GO TO 1137	STEP	34
	IF (IFG.LT.1) GO TO 1137	STEP	35
	IF (IFG.GT.2) GO TO 1137	STEP	36
	IF (IFG.EQ.1) GO TO 1002	STEP	37
	GO TO 5	STEP	38
C	DETERMINE LINEAR AND QUADRATIC COMPONENTS OF PHASE	STEP	39
1137	CALL ILLI(AX,AY,RAOCN,IPS)	STEP	40
	IF (IFG.LT.1) GO TO 1139	STEP	41
	IF (IFG.GT.2) GO TO 1139	STEP	42
	IF (IFG.EQ.1) GO TO 1002	STEP	43
	GO TO 5	STEP	44
1139	RHNEAK=1.E70	STEP	45
	IF (DABS(RAOCUR/OELZ).GT.RHNEAK) GO TO 1002	STEP	46
C	*****	STEP	47
C	VARIABLE AREA MESH PROPAGATION TRANSFORMATION TO EQUIVALENT	STEP	48
C	COLLIMATED BEAM	STEP	49
	5 ALPHA=10.	STEP	50
C	DETERMINATION OF BEAM WAIST AND DISTANCE TO IT	STEP	51
	W1 = ALPHA*UCALC1/2.	STEP	52
	WW = (W1*W1*PI/WL)**2	STEP	53
	Z1 = HAUCUM*WW/(HAUCUM**2+WW)	STEP	54
	WU = USQHI(OSQRT(HAUCUM*Z1-Z1**2)*WL/PI)	STEP	55
	ZHAL = PI*WU*WU/WL	STEP	56
	ANZ=2.	STEP	57
	IF (DABS(Z1).LT.ZHAL/ANZ) NNEG=1	STEP	58
	IF (DABS(Z1).GT.ZHAL*ANZ) NNEG=2	STEP	59
	IF (NNEG.EQ.0) GO TO 12	STEP	60
	IF (DABS(Z1*OELZ).GT.ZHAL/ANZ.AND.NNEG.EQ.1) GO TO 12	STEP	61
	IF (DABS(Z1*OELZ).LT.ZHAL*ANZ.AND.NNEG.EQ.2) GO TO 12	STEP	62
	DUME = W1**2*ZHAL/(UCALC1/W1)**2	STEP	63
	IPNT = 1	STEP	64
C	ESTABLISH PROPAGATING MATRIX	STEP	65
C	INCLUDES FREQUENCY SPACE DATA WINDOW	STEP	66
	UU 101 J=2,NPP1	STEP	67
	AJMISQ = (J-1)**2	STEP	68
	WFACTR = 1.0	STEP	69
	IF (J.GT.N0 .AND. NWNUOK.GT.0)	STEP	70
	1 WFACTR = (1.0-COS(PI*FLOAT(NPP1-J)/FLOAT(NWNUOK)))/2.0	STEP	71
	DU 101 I=1,J	STEP	72
	DUM = (AJMISQ*(I-1)**2)	STEP	73
	IPNT = IPNT+1	STEP	74
	APH(1,IPNT)=WFACTR	STEP	75
101	APR(2,IPNT)=DUME*DUM	STEP	76
	TNZ1 = Z1/ZHAL	STEP	77
	IJI=0	STEP	78
	DU 2 K=1,NPY	STEP	79
	YSQ = X(K)**2	STEP	80
	DU 2 I=1,NPTS	STEP	81
	IJI = IJI + 1	STEP	82
	IJI2 = IJI * 2	STEP	83
	IJI2M1 = IJI2 - 1	STEP	84
	PHI = (X(I)**2 + YSQ)*TNZ1/W1**2	STEP	85
	SINP = SIN(PHI)	STEP	86
	COSP = COS(PHI)	STEP	87
	CUMS = CUR(IJI2M1)	STEP	88
	CUN(IJI2M1) = W1*(CUMS*COSP + CUN(IJI2)*SINP)	STEP	89
2	CUN(IJI2) = W1*(CUMS*SINP + CUN(IJI2)*COSP)	STEP	90
	IF (NWT.NE.0) ZKEEP=ZZZ	STEP	91
	ZZZ = Z1	STEP	92
	ZINTE=0.	STEP	93
	WJ=W1	STEP	94
	IF (IFG.EQ.0) ITH=1	STEP	95
	GO TO 2000	STEP	96
C	*****	STEP	97
C	CONSTANT AREA MESH PROPAGATION	STEP	98
C	INCLUDES FREQUENCY SPACE DATA WINDOW	STEP	99

1002	ACDUM1=2.*PI/WL	STEP	100
	DUM1 = (WL/OCALC1)**2	STEP	101
	IPNT = 1	STEP	102
C	ESTABLISH PROPAGATING MATRIX	STEP	103
	DO 200 J=2,NPP1	STEP	104
	AJM15U = (J-1)**2	STEP	105
	WFACHT = 1.0	STEP	106
	IF (J.GT.40 .AND. N=NDOOK.GT.0)	STEP	107
	1 WFACTR = (1.0-COS(PI*FLOAT(NPP1-J)/FLOAT(N=NDOOK)))/2.0	STEP	108
	DO 200 I=1,J	STEP	109
	DUM = (AJM15U+((-1)**2)	STEP	110
	DUM2 = DUM*UUM	STEP	111
	DUM3 = (0.125*DUM2+0.5)*DUM2	STEP	112
	IPNT = IPNT+1	STEP	113
	APH(1,IPNT)=WFACHT	STEP	114
200	APH(2,IPNT)=ACDUM1*UUM3	STEP	115
C	ENTER ROUTINE HERE WHEN CONTINUING WITH SAME PROPAGATING MATRIX	STEP	116
C	ENTRY COME(UELZ,ITH,NWMT)	STEP	117
2000	ZZZ=ZZZ+UELZ	STEP	118
	IF (NWMT.NE.0) GO TO 402	STEP	119
	ZINTE=ZINTE+UELZ	STEP	120
	ZZIN=ZZINTE+UELZ	STEP	121
	XMESH = X(NPTS)-2.*X(1)*X(2)	STEP	122
	HCEK=(1.-2.*WNOXA)*X(NPTS)	STEP	123
	IF (HAPTH.GE.HCEK)HAPTH=0.	STEP	124
402	IF (HNEG.EQ.0) GO TO 92	STEP	125
	WNOU=W0*USUMT(1.*(ZZZ/ZHAL)**2)	STEP	126
	XAPANU=WNOU/WJ	STEP	127
	WJ=WNOU	STEP	128
C	ADJUST BEAM COORDINATES FOR MAGNIFICATION AND MIRROR TILT	STEP	129
	DO 93 I=1,NPTS	STEP	130
	93 X(I)=X(I)*XAPANU	STEP	131
	92 DXNEAL=DXNEAL* SIN (AX) * UELZ	STEP	132
	DYNEAL=DYNEAL* SIN (AY) * UELZ	STEP	133
	IF (WNOXA.LE.0.0) GO TO 49	STEP	134
C	APPLY X-SPACE COSINE DATA =INUOU	STEP	135
	DO 212 I=1,NPTS	STEP	136
	DO 212 J=1,NWNOXA	STEP	137
	IJ2 = I * (NPTS -J) * NPTS	STEP	138
	IF (NPY.EQ.NPTS) CU(IJ2) = CU(IJ2) * FACHT(J)	STEP	139
	IJI=I+(J-1)*NPTS	STEP	140
212	CU(IJI)=CU(IJI)*FACHT(J)	STEP	141
	DO 213 J=1,NPY	STEP	142
	IJ = (J-1)*NPTS	STEP	143
	DO 213 I=1,NWNOXA	STEP	144
	I2=NPTS+1-I	STEP	145
	CU(I+IJ)=CU(I+IJ)*FACHT(I)	STEP	146
213	CU(I2+IJ)=CU(I2+IJ)*FACHT(I)	STEP	147
C	UNFOLD SYMETHIC FIELD FOR FFT USE	STEP	148
	49 IF (NPTS.EQ.NPY) GO TO 50	STEP	149
	DO 15 J=1,NPY	STEP	150
	DO 15 I=1,NPTS	STEP	151
	IJ = I *NPTS+(J-1)	STEP	152
	IJI = I * (NPTS-J)*NPTS	STEP	153
	15 CU(IJI) = CU(IJ)	STEP	154
C	***** STEHL INTENSITY CALCULATION *****	STEP	155
C	* STEHL INTENSITY IS CALCULATED FROM THE CENTERLINE INTENSITY *	STEP	156
C	* OF THE FAN FIELD DISTRIBUTION. THE METHOD USES THE CENTERLINE *	STEP	157
C	* COEFFICIENT OF THE FFT FOR THE UNNORMALIZED CENTERLINE *	STEP	158
C	* INTENSITY. POWER CONSERVATION IS USED TO DEFINE THE PLANE WAVE *	STEP	159
C	* NEAR FIELD INTENSITY VALUE. THE RATIO OF CENTERLINE INTENSITY *	STEP	160
C	* (FFT) TO PEAK INTENSITY (PLANE WAVE) DEFINES STEHL INTENSITY. *	STEP	161
C	* IN THIS ROUTINE. J FORGHAM 10 28 74 *	STEP	162
C	*****	STEP	163
50	IF (HAPTH.EQ.0.0.OR.NWMT.EQ.1)GO TO 96	STEP	164
	XITOT = 0.	STEP	165
	PI = 3.141596	STEP	166
	XMESH = XMESH**.	STEP	167
	NOB=NPTS*NPTS	STEP	168
	DO 95 I=1,NOB	STEP	169
	I2 = I * 2	STEP	170

	XITOT = XITOT + CUM(12-1)**2 + CUM(12)**2	STEP	171
95	CONTINUE	STEP	172
C	XITOT = INTEGRAL OF INTENSITY (UNNORMALIZED)	STEP	173
C	CUM(1) CONTAINS CENTER LINE FFT OF NEAR FIELD DISTRIBUTION AFTER	STEP	174
C	RETURN FROM "FOUNT".	STEP	175
C	TRANSFORM COMPLEX FIELD TO FREQUENCY SPACE WITH FFT	STEP	176
96	CALL FOUNT(CU,NAH,NUD,1)	STEP	177
	IF (RAPHX.EQ.0.0.UH.NHNT.EQ.1) GO TO 99	STEP	178
	AHEA = PI*RAPHX**2	STEP	179
	AHEASU = AHEA * AHEA	STEP	180
	XIBAH = XITOT / NUH	STEP	181
	XIBHPW = XIBAH * ((XMESM*AMESM)/AHEA)	STEP	182
C ***	XIBHPW = PLANE WAVE INTENSITY (NEAR FIELD)	STEP	183
	NUBSU = NUH * NUH	STEP	184
	XINOHM = AMESM / NUBSU	STEP	185
C	CLIFF = (CUM(1)**2 + CUM(2)**2) * XINOHM	STEP	186
C	CLIFF = CENTERLINE INTENSITY (FAR FIELD)	STEP	187
C	STHEML INTENSITY	STEP	188
	STHINT = CLIFF / (XIBHPW * AHEASU)	STEP	189
	WRITE (6,10) STHINT	STEP	190
10	FORMAT(//2X,19H STHEML INTENSITY = ,012.5)	STEP	191
99	RAPHX=0.0	STEP	192
	DTZ=DELZ	STEP	193
C	CALCULATE DELZ IN EQUIVALENT COLLIMATED COORDINATE SYSTEM	STEP	194
	IF (NNEG.EQ.1) DTZ=UATAN(ZZZ/ZHAL)-UATAN((ZZZ-DELZ)/ZHAL)	STEP	195
	IF (NNEG.EQ.2) DTZ=DELZ/(ZZZ*(ZZZ-DELZ))	STEP	196
	IPNT = 1	STEP	197
	CUM(1)=CUM(1)*ANPZ	STEP	198
C	APPLY PROPAGATION MATRIX	STEP	199
	DO 100 J=2,NPP1	STEP	200
	J1 = NPZ*2-J	STEP	201
	DO 100 I=1,J	STEP	202
	I1 = NPZ*2-I	STEP	203
	IPNT = IPNT+1	STEP	204
	PHI = DTZ * APR(2,IPNT)	STEP	205
	SINP = SIN(PHI)	STEP	206
	COSP = COS(PHI)	STEP	207
	ACNST = ANPZ * APR(1,IPNT)	STEP	208
	CDUMR(1) = ACNST * COSP	STEP	209
	CDUMR(2) = ACNST * SINP	STEP	210
C	CDUM=ANPZ*APR(1,IPNT)*CEXP(CMPLX(0.,APR(2,IPNT)*DTZ))	STEP	211
	CUM(I+NPZ*(J-1)) = CUM(I+NPZ*(J-1))*CDUM	STEP	212
	IF (I.EQ.J) GO TO 108	STEP	213
	CUM(J+NPZ*(I-1)) = CUM(J+NPZ*(I-1))*CDUM	STEP	214
	IF (J.EQ.NPP1) GO TO 109	STEP	215
	CUM(I+NPZ*(J1-1)) = CUM(I+NPZ*(J1-1))*CDUM	STEP	216
	CUM(J1+NPZ*(I1-1)) = CUM(J1+NPZ*(I1-1))*CDUM	STEP	217
	IF (I.LT.2) GO TO 100	STEP	218
	CUM(I1+NPZ*(J-1)) = CUM(I1+NPZ*(J-1))*CDUM	STEP	219
	CUM(J+NPZ*(I1-1)) = CUM(J+NPZ*(I1-1))*CDUM	STEP	220
	CUM(I1+NPZ*(J1-1)) = CUM(I1+NPZ*(J1-1))*CDUM	STEP	221
	CUM(J1+NPZ*(I1-1)) = CUM(J1+NPZ*(I1-1))*CDUM	STEP	222
	GO TO 100	STEP	223
108	IF (I.EQ.NPP1) GO TO 100	STEP	224
	CUM(I+NPZ*(J1-1)) = CUM(I+NPZ*(J1-1))*CDUM	STEP	225
	CUM(J1+NPZ*(I1-1)) = CUM(J1+NPZ*(I1-1))*CDUM	STEP	226
	CUM(I1+NPZ*(J1-1)) = CUM(I1+NPZ*(J1-1))*CDUM	STEP	227
	GO TO 100	STEP	228
109	IF (I.LT.2) GO TO 100	STEP	229
	CUM(I1+NPZ*(J-1)) = CUM(I1+NPZ*(J-1))*CDUM	STEP	230
	CUM(J+NPZ*(I1-1)) = CUM(J+NPZ*(I1-1))*CDUM	STEP	231
100	CONTINUE	STEP	232
C	TRANSFORM COMPLEX FIELD TO X-SPACE WITH FFT	STEP	233
	CALL FOUNT(CU,NAH,NUD,-1)	STEP	234
	IF (NNT.NE.0) ZZZ=ZKEEP	STEP	235
	IF (ITH.EQ.0.UH.NNEG.EQ.0) RETURN	STEP	236
C	TRANSFORM FROM EQUIVALENT COLLIMATED COORDINATE SYSTEM (X,Y)	STEP	237
C	BACK TO NEAR COORDINATE SYSTEM (X,Y).	STEP	238
	WF = WOSURT(1.,(ZZZ/ZHAL)**2)	STEP	239
	TNZZ = ZZZ/ZHAL	STEP	240
	FF=TNZZ*(WF*WF)	STEP	241
	DO 42 J=1,NPT	STEP	242

YSQ = X(J)**2	STEP	243
OU 42 I=1,NPTS	STEP	244
IJI = I*(J-1)*NPTS	STEP	245
IJI2 = 2 * IJI	STEP	246
IJI2M1 = IJI2 - 1	STEP	247
PMI = -(X(I)**2 * YSQ) **FF	STEP	248
SINP = SIN (PMI)	STEP	249
COSP = COS(PMI)	STEP	250
CUMS = CUR(IJI2M1)	STEP	251
CUM(IJI2M1) = (CUMS*COSP - CUM(IJI2)*SINP)/WF	STEP	252
42 CUM(IJI2) = (CUMS*SINP + CUM(IJI2)*COSP)/WF	STEP	253
XXPAND=WF/W1	STEP	254
NREG = 0	STEP	255
WRITE (6,522) XXPAND	STEP	256
522 FORMAT (/37H THE MAGNIFICATION OF THE FIELD IS ,F10.6/)	STEP	257
RETURN	STEP	258
12 WRITE (6,9)	STEP	259
9 FORMAT(///.33H INVALID VARIABLE MESH REGION ,/.53H SUBROUTI	STEP	260
1NE STEP COUNTINUING WITH CONSTANT MESH ,/.65H NOTE POSSIBLE EXP	STEP	261
1ANSION OF THE BEAM OUTSIDE THE CALC. REGION ,///)	STEP	262
IFG=1	STEP	263
NNEG=0	STEP	264
GO TO 1002	STEP	265
END	STEP	266

34. SUBROUTINE TBLOOM

a. Purpose -- This subroutine, shown in Figure 68, is used to model four types of thermal blooming which may be seen by a beam as it propagates through an absorptive medium.

The four types are:

1. Transverse
2. Axial
3. Free convective
4. Transient

b. Relevant formalism -- Thermal blooming arises as a consequence of the absorption of laser radiation by the transmitting gas. The absorbed radiation heats the gas and consequently changes its refractive index. These variations in the index of refraction induce phase changes in the propagated beam. Phase changes produced by thermal blooming can result in beam divergence, which overloads apertures and provides a source of high energy feedback. Thermal blooming also degrades beam quality. Thermal blooming models are available in the SOQ library to describe the impact on the beam phase and amplitude produced when thermal blooming occurs in (1) a transverse flow field, (2) an axial flow field, (3) a free convective flow field, and (4) transient conditions with no external flow.

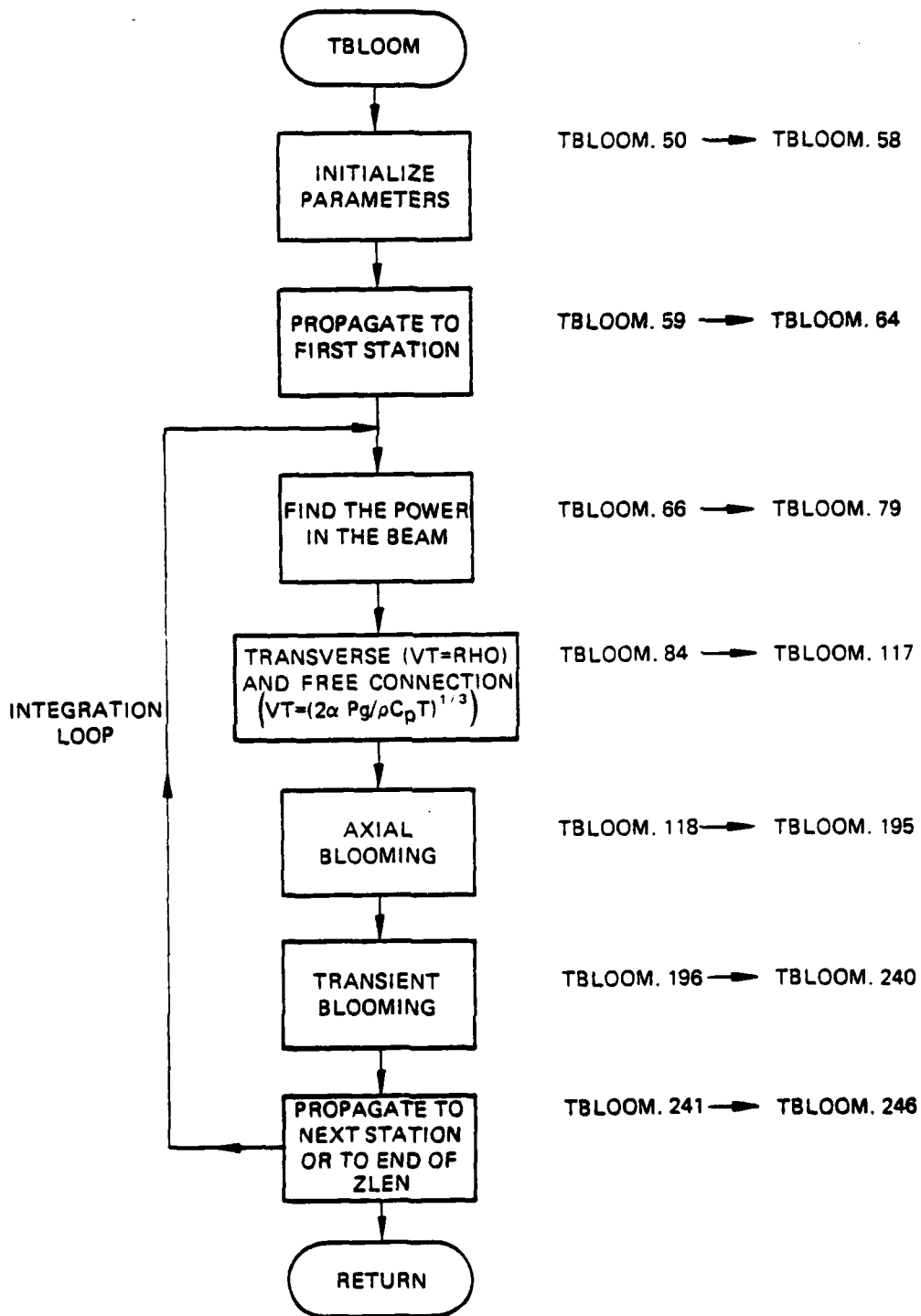


Figure 68. Subroutine TBLOOM flow chart.

Figure 69 schematically demonstrates the procedure used to modify the complex field, $U(x,y)$, as it is propagated through a thermal blooming gain phase segment within the SOQ code.

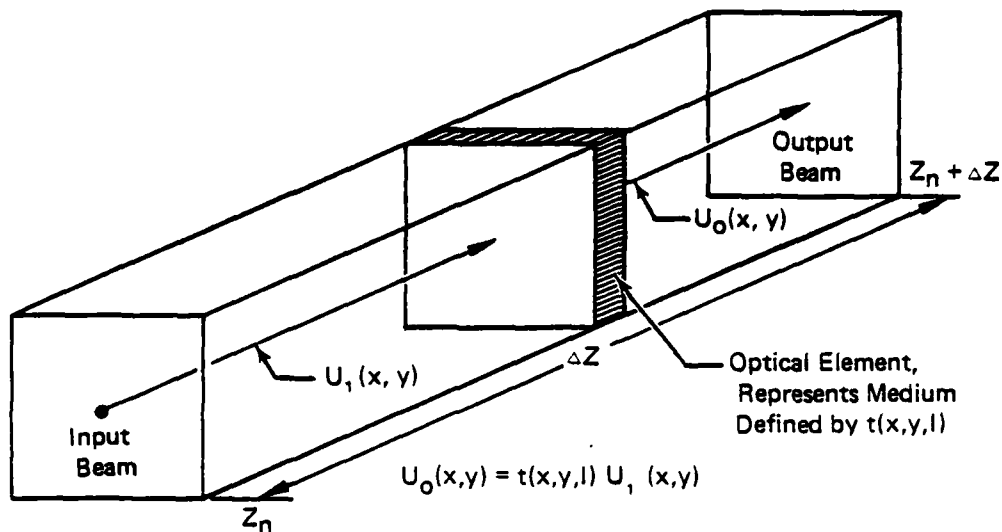


Figure 69. Illustration of thermal blooming model.

As the beam is propagated a distance ΔL through the medium, it is continuously interacting with that medium. By requiring that the effect is small, the integrated effect can be approximated by a finite number of discrete steps in the following manner:

Assume each step is of length ΔL and that the effect of such a step is approximated by a vacuum propagation to the center ($\Delta L/2$), application of the appropriate transmission function $t(x,y,l)$, followed by subsequent vacuum propagation of field the remaining distance ($\Delta L/2$).

The transmission function $t(x,y,\Delta L, I(x,y))$ can be assumed to be of the form

$$t(x,y,l) = \exp\left[\frac{\alpha\Delta L}{2} - i\Delta\phi\right] \quad (224)$$

where α is the absorptivity of the medium and $\Delta\phi$ can be written

$$\Delta\phi = \frac{2\pi}{\lambda} \frac{dn}{dt} \int_0^{\Delta L} \delta T dz' \quad (225)$$

$$\delta T = \delta T(x, y, z)$$

Employing the usual Gladstone-Dale relationship to approximate the index n , ($n = 1 + \rho C$) and the equation of state for an ideal gas ($P = \frac{RT}{M}\rho$), the expression for $\Delta\phi$ becomes (assuming constant pressure)

$$\Delta\phi = \frac{2\pi}{\lambda} \left(\frac{-\rho C}{T} \right) \int_0^{\Delta L} dz \delta T(x, y, z) \quad (226)$$

δT represents the temperature variation across the beam as a result of one of the four types of thermal blooming. It is found in the following manner:

(1) Transverse blooming -- It is assumed that the wind is blowing with speed V_T (con/scan) from the negative x-direction. The resulting temperature variation is:

$$\delta T_T = \frac{\alpha}{\rho C_p V_T} \int_{-\infty}^x I(x', y, z) dx' \quad (227)$$

where I is the intensity of the beam.

(2) Axial blooming -- It is assumed that the wind blows in the same direction the beam is traveling with speed V (cm/sec) resulting in

$$\delta T_{ax} = \frac{\alpha}{\rho C_p V_{ax}} \int_0^x I(x, y, z) dz' \quad (228)$$

(3) Free convection -- The temperature variation due to thermal gradients caused by absorption is:

$$\delta T_c = \frac{\alpha}{\rho C_p V_c} \int_{-\infty}^x I(x', y, z) dx' \quad (229)$$

where

$$V_c = \left(\frac{2\alpha P z g}{\rho C_p T} \right)^{\frac{1}{3}}$$

P(Z') being the total power in the beam at Z' and g, the acceleration due to gravity.

(4) Transient -- Finally, in the process of establishing free convection, the beam has a residence time $T_{(sec)}$ during which the temperature variation is

$$\delta T_{tran} = \frac{\alpha T}{\rho C_p} + I \quad (230)$$

c. Fortran

Argument List

- ALFA - Absorptivity of the medium (cm^{-2})
- CP - Specific heat (J/g-K)
- T - Temperature (K)
- RHO - (1) if $RHO < 1$, it is the density (g/cm^3) used for free convection
(2) if $RHO > 1$, it is the transverse velocity
- ZLEN - Total length of the blooming medium
- NSTEPS - The number of steps required to adequately represent thermal blooming over a distance ZLEN. Phase per step shift usually kept $\leq \frac{\pi}{8}$
- INPT - Flag for intermediate plots
- NPROP - Same as NSTE in cavity
- AXIAL - Axial velocity (cm/sec) and is > 0
- DT - Residency time for transient blooming

None of the above parameters is redefined by this subroutine.

Commons:

The variables in common which are modified are:

- (1) CU: the effect of the blooming is applied to CU
- (2) CFIL: due to its equivalence with the PH and W arrays, it is modified when they are defined.

Computer printouts of subroutine TBLOOM follow.

SUBROUTINE TBLOOM 76/176 OPT=1 FIN 4.6+452 04/27/79 12.25.47

```

SUBROUTINE TBLOOM(ALFA,CP,MMO,ZLEN,NSTEPS,INPT,NPROP,AXIAL,UT) TBLOOM 2
LEVEL 2, CU,CUM,MM,PH TBLOOM 3
COMMON/MELT/CU(16384),CFIL(16512),X(128),L,NPTS,NPY,UNX,UNY TBLOOM 4
COMMON/MAY/WNO,MMEG,MMATH TBLOOM 5
DIMENSION X(16384),PH(16384) TBLOOM 6
REAL CUM(32768) TBLOOM 7
REAL ISAT TBLOOM 8
COMPLEX CU,CFIL TBLOOM 9
EQUIVALENCE (CU(1),CUM(1)) TBLOOM 10
***** TBLOOM 11
C THIS VERSION OF TBLOOM HAS BEEN MODIFIED TO TBLOOM 12
C ACCOMODATE AXIAL BLOOMING CALCULATIONS PER PHASE TBLOOM 13
C TWO-THREE PROPOSAL J PURGHAM 6/75 TBLOOM 14
C ***** TBLOOM 15
C THIS ROUTINE HAS BEEN FURTHER MODIFIED TO ACCOMODATE TRANSIENT TBLOOM 16
C THERMAL BLOOMING CALCULATIONS. TRANSIENT TH.BL. IS THE PHASE TBLOOM 17
C CHANGE WHICH RESULTS FROM ENERGY ADDITION TO THE MEDIUM TBLOOM 18
C WITH NO FORCED OR FREE CONVECTION. WE SOLVE..... TBLOOM 19
C MMO * CP * DTEMP/DTIME = ALFA * (X,Y,Z) TBLOOM 20
C AND FIND PHASE CHANGE FROM THE LINEARIZED INDEX CHANGE... TBLOOM 21
C DELTA N = DN/DTEMP * DELTA TEMP TBLOOM 22
C PURGHAM 12 / 19 / 76 TBLOOM 23
C ***** TBLOOM 24
C EQUIVALENCE (X(1),CFIL(1)),(PH(1),CFIL(8193)) TBLOOM 25
C NST=NPROP TBLOOM 26
C M = 0 TBLOOM 27
C IOUT = 1 TBLOOM 28
C IF (NPROP.EQ.3.OR.NPROP.EQ.5) IOUT = 0 TBLOOM 29
C IF (NPROP.EQ.3) NST=2 TBLOOM 30
C WRITE(6,5) ALFA,CP,T, ZLEN,NSTEPS TBLOOM 31
C 5 FORMAT(119HOFIELD HAS ENTERED SUBSYSTEM TBLOOM = STEADY STATE THER TBLOOM 32
C AXIAL BLOOMING MEDIUM /2 TBLOOM 33
C X5A,25HABSORPTION COEFFICIENT = .612.5.5H CM-1/25X. TBLOOM 34
C X19HSPECIFIC HEAT,CP = .612.5.7H J/GM-K/25X. TBLOOM 35
C X14HTEMPERATURE = .612.5.7H DEG. K/25X. TBLOOM 36
C X12MTHICKNESS = .612.5.3H CM/25X. TBLOOM 37
C X15MNU. ELEMENTS = ,I3) TBLOOM 38
C IF (DT.GT.0.0) GO TO 700 TBLOOM 39
C ***** UT GREATER THAN 0.0 INDICATES TRANSIENT BLUMING ***** TBLOOM 40
C IF (AXIAL .GT. 0.0) WRITE(6,596) AXIAL TBLOOM 41
C 596 FORMAT(25X,18HAXIAL VELOCITY = .612.5. 8H CM/SEC ) TBLOOM 42
C IF (AXIAL .GT. 0.0) GO TO 700 TBLOOM 43
C ***** AXIAL = AXIAL VELOCITY *** TBLOOM 44
C IF (MMO .LT. 1.) WRITE(6,6) MMO TBLOOM 45
C 6 FORMAT(25X,10HNUENSITY = .612.5.7H GM/CM3) TBLOOM 46
C IF (MMO .GT. 1.) WRITE(6,7) MMO TBLOOM 47
C 7 FORMAT(25X,23HTRANSVERSE VELOCITY = .612.5.7H CM/SEC) TBLOOM 48
C 700 DELZ = ZLEN/NSTEPS TBLOOM 49
C GUC = .223 TBLOOM 50
C RAU = 1. TBLOOM 51
C ZLAST = 0. TBLOOM 52
C ZNUM = 0. TBLOOM 53
C AVELAG = 0. TBLOOM 54
C RMSTUT = J. TBLOOM 55
C PHFOT = 0. TBLOOM 56
C PHEJ = EXP(-ALFA*DELZ/2.0) TBLOOM 57
C *** PROPAGATE TO FIRST ELEMENT TBLOOM 58
C IF ( NPROP.GE.4 ) CALL COME(DELZ/2.0,0,M) TBLOOM 59
C IF ( NPROP.GE.4 ) TBLOOM 60
C ICALL STEP(DELZ/2.0,MMO,1.0,1,NST, 0.0,0.0,0.0,M,1) TBLOOM 61
C IF ( NPROP.LE.3 ) TBLOOM 62
C ICALL STEP(DELZ/2.0,MMO,1.0,1,NST, 0.0,0.0,0.0,M,0) TBLOOM 63
C DO 100 K=1,NSTEPS TBLOOM 64
C KMIK=1 TBLOOM 65
C TBLOOM 66

```

DA = X(2) - X(1)	18	TBLOOM	67
UASU = DA**2		TBLOOM	68
DCAL = NPTS*UX		TBLOOM	69
XFACT = 1.		TBLOOM	70
IF(NNEG.EQ.1.OR.NNEG.EQ.2)XFACT = 1./WNW**2		TBLOOM	71
C *** COMPUTE POWER DENSITY		TBLOOM	72
NUH=NPTS*NPY		TBLOOM	73
PT = 0.		TBLOOM	74
DO 10 I=1,NUH		TBLOOM	75
C W(I) = CU(I)*CONJG(CU(I))*XFACT		TBLOOM	76
W(I) = (CU(2*I-1)**2 + CU(2*I)**2) *XFACT		TBLOOM	77
10 PT = PT+W(I)		TBLOOM	78
PT = PT*UASU*NPIS/NPY		TBLOOM	79
IF(DT.GT.0.0) GO TO 220		TBLOOM	80
C *** TEST DT TO DETERMINE IF TRANSIENT BLOOMING REQUIRED		TBLOOM	81
IF (AXIAL .GT. 0.) GO TO 18		TBLOOM	82
C *** TEST AXIAL TO DETERMINE IF AXIAL BLOOMING IS REQUIRED		TBLOOM	83
VT = HHO		TBLOOM	84
IF(RHO .LT. 1.0)		TBLOOM	85
AVT = (980.665*PT*ALFA/(HHO*CP*VT))**(1./3.)		TBLOOM	86
CAPK = 6.2831853*ALFA*DELZ*UCAL/(WL*CP*VT)*GOC		TBLOOM	87
IF(INPT.EQ.0) GO TO 15		TBLOOM	88
IF(MOD(KM1,INPT).NE.0)GO TO 15		TBLOOM	89
WRITE(6,14)K,PT,VT,CAPK		TBLOOM	90
14 FORMAT(40H1 FIELD INCIDENT UPON THERMAL BLOOMING ELEMENT,12,8H POW		TBLOOM	91
1ER= ,G12.5,23H TRANSVERSE VELOCITY = ,G12.5,15HCM/S CAPK = ,G12		TBLOOM	92
1.5)		TBLOOM	93
N = 0		TBLOOM	94
UMAX = 0.		TBLOOM	95
CALL OUTPUT(CU,NPY,NPTS,X,N,UMAX,.TRUE,..FALSE,..FALSE.)		TBLOOM	96
15 PMAX = -1.E7		TBLOOM	97
WAIST2 = 25		TBLOOM	98
19 CONTINUE		TBLOOM	99
DO 20 J=1,NPY		TBLOOM	100
SUM = 0.		TBLOOM	101
J1=(J-1)*NPTS		TBLOOM	102
DO 20 I=1,NPTS		TBLOOM	103
JJ=I+J1		TBLOOM	104
SUM = SUM+W(JJ)		TBLOOM	105
PH(JJ) = CAPK*SUM/NPTS		TBLOOM	106
CU(JJ) = CU(JJ)*CMPLX(COS(PH(JJ)),SIN(PH(JJ)))*PHED		TBLOOM	107
20 IF(PH(JJ).GT.PMAX)PMAX=PH(JJ)		TBLOOM	108
IF(INPT.EQ.0) GO TO 35		TBLOOM	109
IF(MOD(KM1,INPT).NE.0)GO TO 35		TBLOOM	110
WRITE(6,34) K,PMAX		TBLOOM	111
34 FORMAT(54H1 FIELD AFTER MODIFICATION BY THERMAL BLOOMING ELEMENT,1		TBLOOM	112
12,32H MAXIMUM PHASE SHIFT INUUCED WAS,G12.5,8H HAD(ANS)		TBLOOM	113
N = 0		TBLOOM	114
UMAX = 0.		TBLOOM	115
CALL OUTPUT(CU,NPY,NPTS,X,N,UMAX,.TRUE,..FALSE,..FALSE.)		TBLOOM	116
GO TO 35		TBLOOM	117
C *****		TBLOOM	118
C THIS SECTION IS DESIGNED TO CALCULATE PHASE CHANGE OF THE BEAM		TBLOOM	119
C DUE TO AN AXIAL VELOCITY COMPONENT. THE MATH REQUIRES THE SOLEN		TBLOOM	120
C OF THE ENERGY EQUATION FOR A TEMP WISE PARALLEL TO THE BEAM AXIS.		TBLOOM	121
C IN WHAT FOLLOWS, CAPKAX IS A DISTORTION NUMBER OF SPTS, AND		TBLOOM	122
C THE PHASE CHANGE AT EACH MESH POINT RESULTS FROM THE PRODUCT		TBLOOM	123
C OF CAPKAX * INTENSITY "W". THE FIELD IS MODIFIED BY THE PHASE		TBLOOM	124
C CHANGE INUUCED,AND THE POWER LOST TO HEATING THE MEDIUM "PHED".		TBLOOM	125
C *****		TBLOOM	126
18 CAPKAX = 6.2831853*ALFA*GOC / (WL*CP*AXIAL*1*2.)		TBLOOM	127
ZNOW = ZNOW + DELZ		TBLOOM	128
IF (INPT .EQ. 0) GO TO 50		TBLOOM	129
IF(MOD(KM1,INPT).NE.0)GO TO 50		TBLOOM	130
WRITE(6,45)K,PT,AXIAL,CAPKAX		TBLOOM	131
WRITE (6,46) ZNOW		TBLOOM	132
45 FORMAT(40H1 FIELD INCIDENT UPON THERMAL BLOOMING ELEMENT,12,8H POW		TBLOOM	133
1ER= ,G12.5,23H AXIAL VELOCITY = ,G12.5,15HCM/S CAPKAX= ,		TBLOOM	134
2 G12.5)		TBLOOM	135
46 FORMAT(10X,19HAXIAL POSITION = ,G12.5,3H CM)		TBLOOM	136
N = 0		TBLOOM	137
UMAX = 0.		TBLOOM	138

```

CALL OUTPUT(CU,NPY,NPTS,X,N,UMAX,,TRUE,,FALSE,,FALSE.)
C ***** THE DO 200 LOOP IS AN ANALYTICAL GAUSSIAN BLOOM *****
C ***** THE DO 200 ALSO CALCULATES PHASE-GAIN NUMERICALLY *****
50 PMAXX = 1.E+7
EWAIST = 5.0
PMBAR = 0.0
PMSQ = 0.0
DO 200 J = 1,NPY
  J1=(J-1)*NPTS
  DO 200 I = 1,NPTS
    C ANG = X(I) * X(I) + X(J)*X(J)
    C WAISTZ = EWAIST * EWAIST
    C IF (ANG .GE. WAISTZ) ANG = 0.0
    C PHGAUS = CAPKAX * (PT / 3.14159) * (1./WAISTZ) * (EXP((-ANG * 2.)/
    C X WAISTZ)) * 2.31*(ZNOW**2 - ZLAST**2)
    C KK = I + J1
    C PH(KK) = CAPKAX * W(KK)*(ZNOW**2 - ZLAST**2)
    C CU(KK) = CU(KK) + CMPLX(COS(PH(KK)),SIN(PH(KK))) * PHED
    C DELTA = PHGAUS - PH(KK)
    C PMBAR = PMBAR + PH(KK)
    C PMSQ = PMSQ + PH(KK) * PH(KK)
    C IF (J .NE. 1 + NPY/2) GO TO 181
    C IF (INPT .EQ. 0) GO TO 179B
    C WRITE (6,180)X(I),X(J),PHGAUS,PH(KK),DELTA
C179B CONTINUE
C 180 FORMAT(5X,5G12.5)
C 181 CONTINUE
200 IF (PH(KK) .GT. PMAXX) PMAXX = PH(KK)
C *****
C RMSPHS = RMS PHASE DISTORTION FOR DELZ STEP
C AVELAG = AVERAGE PHASE LAG FOR THERMAL BLOOMING SEGMENT
C PMBANI = AVERAGE PHASE LAG FOR DELZ STEP
C RMSTOT = TOTAL RMS PHASE FOR THERMAL BLOOMING SEGMENT
C PHTOT = TOTAL MAXIMUM PHASE LAG FOR THERMAL BLOOMING SEGMENT
C THE ABOVE STATISTICAL PARAMETERS ARE INCLUDED AS DIAGNOSTICS
C ***** JULY 8/26/74 *****
C RMSPHS = SQRT(PMSQ - ((PMBAR**2)/(NPY*NPTS)))
C TOTPTS = NPY * NPTS
C RMSPHS = RMSPHS / SQRT(TOTPTS)
C PMBANI = PMBAR / (NPY*NPTS)
C AVELAG = AVELAG + PMBANI
C RMSTOT = SQRT(RMSTOT**2 + RMSPHS**2)
C PHTOT = PHTOT + PMAXX
ZLAST = ZNOW
IF (INPT .EQ. 0) GO TO 35
IF (MOD(KM1,INPT).NE.0)GO TO 35
WRITE (6,33) K,PMAXX,AXIAL
C WRITE (6,49) AVELAG,RMSTOT,PHTOT
33 FORMAT (22H1 FIELD AFTER AXIAL TB,12.5,3MPHAX,=,G12.5,4MVAX,=,G12.5)
C 49 FORMAT(5X,24MTOTAL AVERAGE PHASE LAG ,G12.5,18H TOTAL RMS PHASE = ,
C 12.5,26H TOTAL PHASE CHANGE MAX. =,G12.5)
WRITE (6,44) CAPKAX
44 FORMAT(10X,10H CAPKAX = ,G12.5)
N = 0
UMAX = 0.
CALL OUTPUT(CU,NPY,NPTS,X,N,UMAX,,TRUE,,FALSE,,FALSE.)
GO TO 35
C *****
C TRANSIENT THERMAL BLOOMING CALCULATIONS ARE DONE IN THIS SECTION.
C ENERGY EQUATION IS SOLVED FOR PHASE CHANGE AS A FUNCTION OF
C BEAM ON TIME.
C *****
220 ETA = (ALFA * GDC) / (1 + CP)
ZNOW = ZLAST + DELZ
IF (INPT .EQ. 0) GO TO 210
IF (MOD(KM1,INPT).NE.0)GO TO 210
WRITE(6,274) OT,ETA,DELZ,ZNOW,AFAC,PHED
274 FORMAT (7,7H OT = ,G12.5,7H ETA = ,G12.5,8H DELZ = ,G12.5,/,
15H Z = ,G12.5,10H AFAC = ,G12.5,9H PHED = ,G12.5)
WRITE (6,278) K
278 FORMAT(54H1 FIELD INCIDENT ON TRANSIENT THERMAL BLOOMING ELEMENT
X,12)

```

```

TBLOOM 139
TBLOOM 140
TBLOOM 141
TBLOOM 142
TBLOOM 143
TBLOOM 144
TBLOOM 145
TBLOOM 146
TBLOOM 147
TBLOOM 148
TBLOOM 149
TBLOOM 150
TBLOOM 151
TBLOOM 152
TBLOOM 153
TBLOOM 154
TBLOOM 155
TBLOOM 156
TBLOOM 157
TBLOOM 158
TBLOOM 159
TBLOOM 160
TBLOOM 161
TBLOOM 162
TBLOOM 163
TBLOOM 164
TBLOOM 165
TBLOOM 166
TBLOOM 167
TBLOOM 168
TBLOOM 169
TBLOOM 170
TBLOOM 171
TBLOOM 172
TBLOOM 173
TBLOOM 174
TBLOOM 175
TBLOOM 176
TBLOOM 177
TBLOOM 178
TBLOOM 179
TBLOOM 180
TBLOOM 181
TBLOOM 182
TBLOOM 183
TBLOOM 184
TBLOOM 185
TBLOOM 186
TBLOOM 187
TBLOOM 188
TBLOOM 189
TBLOOM 190
TBLOOM 191
TBLOOM 192
TBLOOM 193
TBLOOM 194
TBLOOM 195
TBLOOM 196
TBLOOM 197
TBLOOM 198
TBLOOM 199
TBLOOM 200
TBLOOM 201
TBLOOM 202
TBLOOM 203
TBLOOM 204
TBLOOM 205
TBLOOM 206
TBLOOM 207
TBLOOM 208
TBLOOM 209
TBLOOM 210

```

N = 0	TBLOOM	211
UMAX = 0.0	TBLOOM	212
CALL OUTPUT(CU,NPY,NPTS,X,N,UMAX,,TRUE,,FALSE,,FALSE.)	TBLOOM	213
210 NWRITE = NPY / 2.	TBLOOM	214
ZLAST = ZNOW	TBLOOM	215
POWER = 0.0	TBLOOM	216
FACTON = ETA * DT * UELZ * 6.2831853 / WL * XFACT	TBLOOM	217
DO 300 L = 1, NPY	TBLOOM	218
J = NPTS*(L-1)	TBLOOM	219
DO 300 I = 1, NPTS	TBLOOM	220
IJ = I * J	TBLOOM	221
C XIAY = CU(IJ) * CONJG(CU(IJ))	TBLOOM	222
XIAY = CUM(2*IJ-1)**2 + CUM(2*IJ)**2	TBLOOM	223
DPHI = FACTON * XIAY	TBLOOM	224
C DPHI = (ETA * DT * UELZ * 6.2831853 / WL) * XIAY	TBLOOM	225
C CU(IJ) = CU(IJ) * CEXP(CMPLX(U., DPHI)) * PRED	TBLOOM	226
CU(IJ) = CU(IJ) * CMPLX(COS(DPHI), SIN(DPHI)) * PRED	TBLOOM	227
C 300 POWER = POWER + CU(IJ)*CONJG(CU(IJ))	TBLOOM	228
300 POWER = POWER * XIAY	TBLOOM	229
POWER = POWER * DASD * NPTS/NPY * XFACT	TBLOOM	230
WRITE(6,295) PT,POWER	TBLOOM	231
295 FORMAT(10X,6M PT = ,G12.5,10M POWER = ,G12.5)	TBLOOM	232
IF (INPT .EQ. 0) GO TO 35	TBLOOM	233
IF (MOD(KM1,INPT).NE.0)GO TO 35	TBLOOM	234
WRITE (6,281) K	TBLOOM	235
281 FORMAT(49M1 FIELD AFTER TRANSIENT THERMAL BLOOMING SEGMENT ,I2)	TBLOOM	236
UMAX = 0.	TBLOOM	237
N=0	TBLOOM	238
CALL OUTPUT (CU,NPY,NPTS,X,N,UMAX,,TRUE,,FALSE,,FALSE.)	TBLOOM	239
250 CONTINUE	TBLOOM	240
C 35 IF (K.LT.NSTEPS) CALL CUME(UDELZ,0,M)	TBLOOM	241
35 IF (K.LT.NSTEPS)	TBLOOM	242
ICALL STEP(UDELZ ,MAU,.1,.1,NST, 0,0,0,0,0,M,1)	TBLOOM	243
C 100 IF (K.EQ.NSTEPS) CALL CUME(UDELZ/2.,10UT,M)	TBLOOM	244
100 IF (K.EQ.NSTEPS)	TBLOOM	245
ICALL STEP(UDELZ/2.,MAU,.1,.1,NST,10UT,0,0,0,0,M,1)	TBLOOM	246
RETURN	TBLOOM	247
END	TBLOOM	248

35. SUBROUTINE THERML

a. Purpose -- Since uncooled mirror glass has such a low coefficient of thermal expansion, the mirror surface heats up as the beam hits it, thus heating up the surrounding boundary layer of air. Subroutine THERML, shown in Figure 70, models the phase change impressed on the beam due to thermal gradients in the boundary layer of air.

b. Relevant formalism -- The theory of this phenomenon was developed by Humphreys and Wick (Ref. 15) of AFWL.

15. Humphreys, W. W. and R. V. Wick, "Change in Optical Path Length Near a Hot Mirror Surface," Laser Digest, AFWL-TR-75-140, 1975, p. 9.

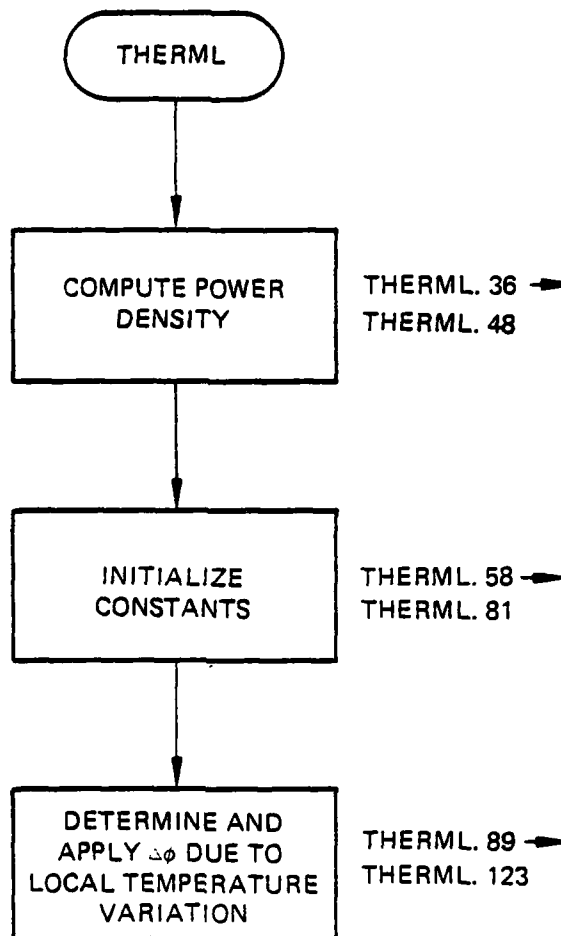


Figure 70. Subroutine THERML organization.

Following Humphreys and Wick, assume that the times of interest are short enough to consider the mirror to be a semi-infinite slab. From the theory of heat conduction the time for heat to traverse a length L is $t = L^2/\alpha$. Thus, for mirrors of thickness L , the time during which the mirror acts like a semi-infinite slab is $\ll L^2/\alpha$. Assume also that for these times one can neglect natural convective cooling. Therefore, the air can also be modeled as a semi-infinite slab. The one-dimensional heat equation is then assumed to apply for both the mirror and the air:

$$\frac{\partial^2 T_m}{\partial x_m^2} = \frac{1}{\alpha_m} \left(\frac{\partial T_m}{\partial t} \right) \quad \frac{\partial^2 T_a}{\partial x_a^2} = \frac{1}{\alpha_a} \left(\frac{\partial T_a}{\partial t} \right) \quad (231)$$

Common variable altered:

CU = the field is modified by the boundary layer
temperature gradients.

Subroutines called: OUTPUT

where the coordinates are seen in Figure 71.

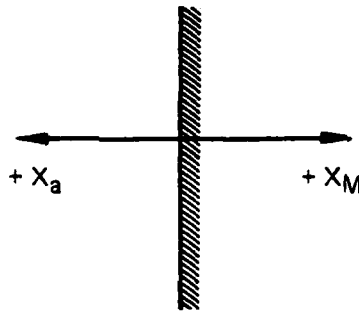


Figure 71. One-dimension heat diagram of mirror and air.

Initially, both the air and the mirror are at the same temperature T_0

$$T_m(x_m, 0) = T_0 = T_a(x_a, 0) \quad (232)$$

For the times considered, the heat does not have time to diffuse to the back boundary of either the mirror or the air. This boundary condition can be written

$$T_m(\infty, t) = T_0 = T_a(\infty, t) \quad (233)$$

The air and the mirror are assumed to maintain the same temperature at their joint boundary so

$$T_m(0, t) = T_a(0, t) \quad (234)$$

The remaining condition to be applied is that of heat balance at the joint boundary. By Fourier's law

$$-k_m \left. \frac{\partial T_m}{\partial x_m} \right|_{x_m = 0} = \alpha I \quad (235)$$

where α is the absorptivity of the mirror. Similarly using Fourier's law at the air boundary

$$-k_a \left. \frac{\partial T_a}{\partial x_a} \right|_{x_a = 0} \quad (236)$$

By combining these two equations, the joint heat balance equation at the boundary becomes:

$$-k_m \left. \frac{\partial T_m}{\partial x_m} \right|_{x_m = 0} - k_a \left. \frac{\partial T_a}{\partial x_a} \right|_{x_a = 0} = \alpha I \quad (237)$$

Since both the media obey the same form of equation, consider the solution of the following equation:

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (238)$$

Finding the Laplace Transform of the above equation gives

$$\frac{d^2 \bar{T}(x,s)}{dx^2} = \frac{1}{\alpha} \left(-T(x,0) + s \bar{T}(x,s) \right) \quad (239)$$

where,

$$\bar{T}(x,s) = \int_0^{\infty} dt e^{-st} T(x,t)$$

Noting that $T(x,0) = T_0$ for both the mirror and the boundary layer, one can rewrite this as

$$\frac{d^2}{dx^2} \left(\bar{T}(x,s) - \frac{T_0}{s} \right) = \frac{s}{\alpha} \left(\bar{T}(x,s) - \frac{T_0}{s} \right) \quad (240)$$

which integrates to give

$$\bar{T}(x,s) = \frac{T_0}{s} + A(s) \epsilon^{\sqrt{\frac{s}{\alpha}} x} + B(s) \epsilon^{-\sqrt{\frac{s}{\alpha}} x} \quad (241)$$

The boundary condition for $x \rightarrow \infty$ implies that $A = 0$ for both media.

Therefore

$$\bar{T}(x,s) - \frac{T_0}{s} = B(s) \epsilon^{-\sqrt{\frac{s}{\alpha}} x} \quad (242)$$

To proceed further, it is necessary to determine $B(s)$. This is done using the joint boundary conditions. Recall that

$$-k_m \left. \frac{\partial T_m}{\partial x_m} \right|_{x_m=0} = -k_a \left. \frac{\partial T_a}{\partial x_a} \right|_{x_a=0} = \alpha I$$

Assuming (αI) to be constant in time, this transforms to

$$-k_m \left. \frac{\partial \bar{T}_m}{\partial x_m} \right|_{x_m=0} = -k_a \left. \frac{\partial \bar{T}_a}{\partial x_a} \right|_{x_a=0} = \frac{\alpha I}{s} \quad (243)$$

but

$$\left. \frac{\partial \bar{T}}{\partial x} (x,s) \right|_{x=0} = -\sqrt{\frac{s}{\alpha}} \epsilon^{-\sqrt{\frac{s}{\alpha}} x} B(s) \Big|_{x=0} = -\sqrt{\frac{s}{\alpha}} B(s) \quad (244)$$

Therefore

$$-k_m \left(-\sqrt{\frac{s}{\alpha}} B_m \right) - k_a \left(-\sqrt{\frac{s}{\alpha}} B_a \right) = \frac{\alpha I}{s} \quad (245)$$

Recall that at $x = 0$, $T_m(0,t) = T_a(0,t)$. This implies that $B_m(s) = B_a(s)$.

Therefore

$$B_a = B_m = \frac{\alpha I}{s\sqrt{s}} \frac{I}{\frac{k_m}{\sqrt{\alpha_m}} + \frac{h_a}{\sqrt{\alpha_a}}} \quad (246)$$

The equation for the air to be back-transformed is therefore

$$\bar{T}_a(x_a, s) - \frac{T_0}{s} = \frac{\alpha I}{\frac{k_m}{\sqrt{\alpha_m}} + \frac{k_a}{\sqrt{\alpha_a}}} \frac{e^{-s \left(\frac{x_a}{\sqrt{\alpha_a}} \right)}}{s} \quad (247)$$

Note that $\bar{T}_m(x_m, t)$ obeys a similar equation with the a and the m subscripts interchanged. Recall the following Laplace Transform theorems:

$$L(T_0) = \frac{T_0}{s}$$

$$\frac{1}{s} L(f(t)) = L \left(\int_0^t dt f(t) \right) \quad (248)$$

and

$$\frac{e^{-a\sqrt{s}}}{\sqrt{s}} = L \left(\frac{e^{-a^2/4t}}{\sqrt{\pi t}} \right) \quad (249)$$

The equation for $T_a(x_a, t)$ is therefore

$$T_a(x_a, t) - T_0 = \frac{\alpha I}{\frac{km}{\sqrt{\alpha_m}} + \frac{ka}{\sqrt{\alpha_a}}} \int_0^t \frac{e^{-\frac{x_a^2}{4\alpha_a t'}}}{\sqrt{\pi t'}} dt' \quad (250)$$

or

$$\begin{aligned} \Delta T_a(x_a, t) &= T_a(x_a, t) - T_0 \\ &= \frac{\alpha I}{\frac{km}{\sqrt{\alpha_m}} + \frac{ka}{\sqrt{\alpha_a}}} 2\sqrt{\frac{t}{\pi}} e^{-\frac{x_a^2}{4\alpha_a t}} \end{aligned} \quad (251)$$

$$- \frac{x_a}{\sqrt{\alpha_a}} \operatorname{erfc} \left(\frac{x_a}{2\sqrt{\alpha_a t}} \right)$$

The phase change in the beam induced by this variation in temperature is given by

$$\Delta \phi(x, y, I) = 2 \left(\frac{2\pi}{\lambda} \right) \int_0^{4\sqrt{\alpha_a t}} \left(\frac{dn}{dT_a} \right) \Delta T_a(dx_a) \quad (252)$$

The factor of 2 is due to the fact that the beam passes through the boundary layer twice. The limit on the integral is seen to be the practical point at which the variation in temperature becomes negligible. This limit is important to estimate since the integral is to be done numerically.

As in TBLOOM, dn/dt is found by the Gladstone-Dale law

$$N = 1 + \rho C \quad (253)$$

and the equation of state of a perfect gas

$$\rho = \frac{MP}{RT} \quad (254)$$

at constant pressure

$$\frac{dn}{dt} = \frac{-\rho C}{T} \quad (255)$$

It is assumed that the effect is small enough that the integral may be approximated by a finite number of steps. Four steps are chosen here.

c. Fortran

Argument List

- CONMIR = mirror thermal conductivity
- CONGAS = boundary layer thermal conductivity
- ALPHAM = mirror diffusivity
- ALPHAG = boundary layer diffusivity
- RHOGAS = boundary layer density
- REFMIR = mirror reflectivity
- TAU = transient time
- TIN = temperature

SUBROUTINE THERML 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE THERML(CONMIR,ALPHAM,ALPHAG,RHOGAS,TAU,TIN,REFMIR,
(CONGAS)
LEVEL 2, CU, CUR
COMMON/MELT/CU(1038*),CFIL(10512),X(128),ML,NPTS,NPY,UNX,UNY
COMMON/WAY/WNOX,INHEG,MAPIN
REAL CU(132708)
COMPLEX CU,CFIL
EQUIVALENCE (CU(1),CU(1))
*****
C          E=1.2/402A,THEM2
C          THIS ROUTINE CALCULATES THE EFFECT OF A THERMAL BOUNDARY
C          LAYER IN FRONT OF A MIRROR. J. FUNGHAM 5/31/75
C          *****
C
C          THIS VERSION CALCULATES PHASE CHANGE BASED ON THE GAS TEMP.
C          RISE IN FRONT OF THE MIRROR ACCORDING TO FORGHAME'S SOLUTION
C          AS GENERATED FROM A HEAT TRANSFER B BY HOLMAN.
C
C          FUNGHAM 5/31/75
C          *****
THERML 2
THERML 3
THERML 4
THERML 5
THERML 6
THERML 7
THERML 8
THERML 9
THERML 10
THERML 11
THERML 12
THERML 13
THERML 14
THERML 15
THERML 16
THERML 17
THERML 18
THERML 19
THERML 20
THERML 21
THERML 22
THERML 23
THERML 24
```

```

WRITE(6,5) ALPHAM,CONMIN,ALPHAG,CUNGA$,RHOGAS,TAU,TIN,HEFMIN THERML 25
5 FORMAT(119HOFIELD HAS ENTERED MINROW THERMAL BOUNDARY LAYER ROUTIN THERML 26
XE. MEDIUM CONDITIONS /2 THERML 27
X5X,30MMINROW DIFFUSIVITY = .G12.5,11M CMSQ/SEC /25X. THERML 28
X30MMINROW THERMAL CONDUCTIVITY = .G12.5,13M WATT/CM SEC /25X. THERML 29
X30MBOUNDARY LAYER DIFFUSIVITY = .G12.5,13M CMSQ/SEC /25X. THERML 30
X30MBOUNDARY LAYER THERMAL CONDUCTIVITY = .G12.5,13M WATT/CM SEC /25X THERML 31
X30MBOUNDARY LAYER DENSITY = .G12.5,9M GM/CC /25X. THERML 32
X19MTRANSIENT TIME = .G12.5,7M SEC /25X. THERML 33
X14MTEMPERATURE = .G12.5,7M DEG. K /25X. THERML 34
X14MMINROW REF. = .G12.5) THERML 35
C *** COMPUTE POWER DENSITY THERML 36
INPT = 1 THERML 37
IF (NPTS.GT.32) INPT = 0 THERML 38
DX = X(2) -X(1) THERML 39
DXXQ = DX * DX THERML 40
XFACT = 1. THERML 41
IF (NNEG .EQ. 1.OR.NNEG .EQ.2) XFACT = 1./4N0W**2 THERML 42
N0W=NPTS*NPY THERML 43
PT = 0. THERML 44
DO 10 I=1,N0W THERML 45
C 10 PT = PT + CU( I ) *CUNJG(CU( I )) *XFACT THERML 46
10 HT = PT + (CUR(2*I-1)**2 + CUR(2*I)**2) * XFACT THERML 47
PT = PT *DXXQ *NPTS /NPY THERML 48
WRITE(6,14) PT THERML 49
14 FORMAT(40M) FIELD INCIDENT UPON BOUNDARY LAYER ELEMENT. 7HPOWER THERML 50
I = .G12.5) THERML 51
IF (INPT .EQ. 0) GO TO 15 THERML 52
N = 0 THERML 53
UMAX = 0. THERML 54
CALL OUTPUT(CU,NPY,NPTS,X,N,UMAX) THERML 55
***** THERML 56
C 15 CONTINUE THERML 57
C *** INITIALIZE CONSTANTS *** THERML 58
C *** ALPHAG THERMAL DIFFUSIVITY OF GAS IN BUY LAYER THERML 59
C *** ALPHAM THERMAL DIFFUSIVITY OF MINROW MATERIAL THERML 60
PI = 3.14159 THERML 61
GUC = .223 THERML 62
EABS = (1. - HEFMIN) THERML 63
WN = (2. * PI) /WL THERML 64
NZ = * THERML 65
NZ1 = NZ * 1 THERML 66
DZ = 1 * . * SQRT(ALPHAG * (AU) ) /NZ THERML 67
SALFA = SQRT(ALPHAG) THERML 68
SALFM = SQRT(ALPHAM) THERML 69
C1 = 1. / (4. * ALPHAG * TAU) THERML 70
C2 = SQRT(C1) THERML 71
C3 = 2. * SQRT( TAU / PI ) THERML 72
C4 = (EABS / CONMIN) * SQRT(PI * ALPHAM * TAU) THERML 73
BIGPHI = -10000. THERML 74
WRITE(6,2192) SALFM,SALFA,CUNGA$,DZ,C1,C2,C3,C4 THERML 75
2192 FORMAT(1UX,2JM SALFM SALFA CUNGA$ DZ .G12.5,/,10X,12MC1 C2 C3 C4 THERML 76
I .G12.5) THERML 77
WRITE(6,1002) EABS,WN THERML 78
1002 FORMAT(1UX,14MMINROW ABS = .G12.5,11M WAVE NO = .G12.5) THERML 79
C *** FIND DM / DTEMP *** THERML 80
DUNIT = (-RHOGAS / TIN) * GUC THERML 81
WRITE(6,1004) DUNIT THERML 82
1004 FORMAT(1UX,9M DUNIT = .G12.5) THERML 83
IF (INPT.EQ.0) GO TO 1014 THERML 84
WRITE(6,1005) THERML 85
1014 CONTINUE THERML 86
1005 FORMAT(1UX,52M X Y DPHIXY THERML 87
X) THERML 88
C *** FIND LOCAL TEMPERATURE AND MODIFY FIELD BY THERMAL LENS *** THERML 89
IJ = 0 THERML 90
DO 400 K = 1,NPY THERML 91
J = (K - 1) * NPTS THERML 92
YY=(K-1) * DX * DX/2. THERML 93
DO 400 I = 1,NPTS THERML 94
TUTN = 0.0 THERML 95
IJ = I + J THERML 96
XX=(I-1) * DX * DX/2. THERML 97

```

C	XIAY = CU(IJ) * CONJG(CU(IJ))	THERML	98
	XIXY = CUR(2*IJ-1)**2 * CUM(2*IJ)**2	THERML	99
	DU 325 MM = 1.0Z1	THERML	100
	ZBL=(MM - 1)*0Z	THERML	101
	ANG1 = -C1 * ZBL * ZBL	THERML	102
	ANG2 = C2 * ZBL	THERML	103
	F2 = ERFC(ANG2)	THERML	104
	DELTA = XIAY * C4 * F2	THERML	105
	TOTN = TOTN + DELTA*UNOT*0Z	THERML	106
325	OPHIXY =-TOTN * #N *2.	THERML	107
C	IF(INPT .EQ. 0) GO TO 330	THERML	108
	IF(OPHIXY.LT.BIGPHI) GO TO 330	THERML	109
	BIGPHI = OPHIXY	THERML	110
	XIMAX = XIXY	THERML	111
	XMAX = XX	THERML	112
	YMAX = YY	THERML	113
	DELTMX =DELTA	THERML	114
C	F1MX =F1	THERML	115
	F2MX =F2	THERML	116
	TUTNMX = TUTN	THERML	117
330	CONTINUE	THERML	118
	IF(INPT .EQ. 0.OR.NPTS .GT.32) GO TO 395	THERML	119
	WRITE(6,1006)XX,YY,OPHIXY	THERML	120
1006	FORMAT(10X,3(10X,G12.5))	THERML	121
C 395	CU(IJ) = CU(IJ) * CEXP(CMPLX(0.,OPHIXY))	THERML	122
395	CU(IJ) = CU(IJ) * CMPLX(COS(OPHIXY),SIN(OPHIXY))	THERML	123
400	CONTINUE	THERML	124
C	IF(INPT.EQ.0) GO TO 35	THERML	125
	WRITE(6,2913) BIGPHI,XIMAX,XMAX,YMAX,DELTMX,F2MX,TUTNMX	THERML	126
2913	FORMAT(10X,14H OPHI,1MAX,X,Y,4G12.5,/,18H DTEMP,F1,F2,DELTA ,3G12.	THERML	127
	15)	THERML	128
	IF(INPT.EQ.0) GO TO 35	THERML	129
34	FORMAT(61H) FIELD AFTER MODIFICATION BY THERMAL BOUNDARY LAYER ELE	THERML	130
	MENT)	THERML	131
	N = 0	THERML	132
	UMAX = 0.	THERML	133
	CALL OUTPUT(CU,NPY,NPTS,X,N,UMAX)	THERML	134
35	RETURN	THERML	135
	END	THERML	136

36. SUBROUTINE TILT

a. Purpose -- Subroutine TILT, shown in Figure 72, can be used to remove beam tilt and will calculate the radius of curvature of a beam.

b. Relevant formalism -- To remove small amounts of beam tilt, the following formalism is used. Large fixed tilts, such as result from mirrors set at an angle to the beam axis, are removed by the system analyst in defining the equivalent collimated system.

Consider an input field $U(x,y)$ incident on an optical element with transmission function $t(x,y)$ yielding an output $U'(x,y)$.

$$\begin{aligned}
 U'(x,y) &= t(x,y) U(x,y) \\
 &= A \exp(i\phi)
 \end{aligned}
 \tag{256}$$

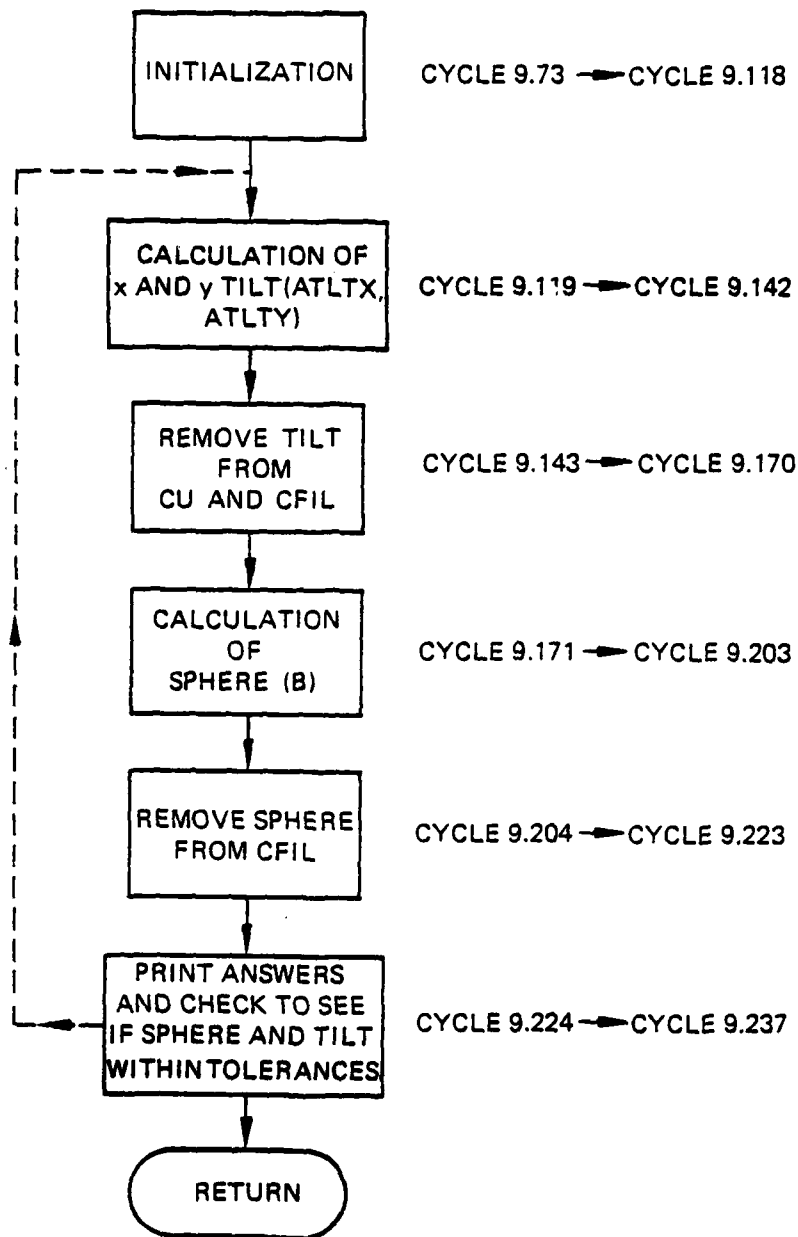


Figure 72. Subroutine TILT organization.

For removal of beam tilt from a field $U(x,y)$ the transmission function must be of the form

$$t_{\text{TILT}}(x,y) = e^{-i(a_x x + a_y y)} = e^{-i\vec{a} \cdot \vec{x}} \quad (257)$$

where $a_x = h\theta_x$ and $a_y = h\theta_y$ define the tilt angles to be removed.

Similarly, the phase curvature is removed by the following transmission function

$$t_{\text{SPHERE}}(x,y) = e^{-i \frac{k}{2R} (x^2 + y^2)} \quad (258)$$

To calculate the constants a_x and a_y for an arbitrary field distribution, $U(x,y)$, define the following functional to be minimized:

$$F_{\text{TILT}} = \iint dx dy |U(x,y)|^2 \left[\nabla(\phi - a_x x - a_y y) \right]^2 \quad (259)$$

or

$$F_{\text{TILT}} = \iint dx dy |U(x,y)|^2 \left[\left(\frac{\partial \phi}{\partial x} - a_x \right)^2 + \left(\frac{\partial \phi}{\partial y} - a_y \right)^2 \right]$$

the resulting expression for \vec{a} is

$$\vec{a} = \langle \vec{\nabla} \phi \rangle \quad (260)$$

where,

$$\langle \vec{\nabla} \phi \rangle = \frac{\iint dx |U(\vec{x})|^2 \vec{\nabla} \phi}{\iint dx |U(\vec{x})|^2} \quad (261)$$

$\nabla \phi$ is easily found from the field data by noting that

$$\text{Im}(U^* \nabla U) = |U|^2 \vec{\nabla} \phi \quad (262)$$

Once the tilt is removed, a similar procedure to remove phase curvature is used. Recall that the transmission function $t_{\text{SPHERE}}(x,y)$ needed is of the form

$$t_{\text{SPHERE}}(x,y) = \epsilon^{-ik\left(\frac{x^2+y^2}{2R}\right)} \quad (263)$$

The new functional to be minimized is

$$F_{\text{SPHERE}} = \iint dx dy \left| U(x,y) \right|^2 \left[\nabla \left(\phi - b \left(\frac{x^2 + y^2}{2} \right) \right) \right]^2 \quad (264)$$

which results in

$$b = \frac{\langle \vec{x} \cdot \vec{\nabla} \phi \rangle}{\langle \vec{x} \cdot \vec{x} \rangle} \quad (265)$$

Values of tilt a and sphere b are found by an iterative procedure until the values established for these parameters do not change appreciably.

c. Fortran

Argument List

$\left. \begin{array}{l} AX \\ AY \end{array} \right\}$ = Total x and y tilt in the beam. The amount of tilt removed from the beam by this routine is added to these parameters so that no tilt information is lost.

RADCUR = the negative of the radius of curvature of the beam found by this routine. To produce a "flat" beam the following calculation would be performed.

$$CU'(I,J) = CU(I,J) * \exp i(\pi/\lambda R) (x^2 + y^2) \quad (266)$$

with R representing RADCUR

$X = X(I)$ and $Y = X(J)$

IPS = the parameter that indicates which options in this routine are to be used. IPS is the same parameter as IIPS in name list PROPGT in subroutine GDL. The options are:

- IPS = 0 Tilt is not called for
- = 1 Tilt only is removed
- = 2 Sphere only found
- = 3 Both tilt and sphere found, tilt being removed.

Common Variables Altered

CU - has tilt removed

CFIL - starts off set to CU, then has both tilt and sphere removed.

Subroutine TILT computer printouts follow.

SUBROUTINE TILT 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.17

	SUBROUTINE TILT(AX,AY,HAUCUM,IPS)	CYCLE9	73
C	PHASE CONNECTION ROUTINE	CYCLE9	74
C	THIS ROUTINE DETERMINES THE LINEAR AND QUADRATIC COMPONENTS OF	CYCLE9	75
C	PHASE. IT ALSO REMOVES THE LINEAR COMPONENT BEFORE RETURNING	CYCLE9	76
C	TO THE CALLING ROUTINE.	CYCLE9	77
	LEVEL 2. CU,CUM,CFILM	CYCLE9	78
	COMMON /MELT/ CU(1638*),CFIL(129,128),X(128),WL,NPTS,NPY,URX,URY	CYCLE9	79
	COMPLEX CU,CFIL,CSUMX,CSUMY,CB,CA,CAA,CCC,CC,CCNJ,CAX,CAY,CFACT,	CYCLE9	80
	A CXX,CHY,CCNJ,CHINT	CYCLE9	81
	DIMENSION CUM(1),CFILR(258,1)	CYCLE9	82
	EQUIVALENCE (CU(1),CUM(1)) , (CFIL(1,1),CFILM(1,1))	CYCLE9	83
	WRITE(6,301)	CYCLE9	84
301	FORMAT(70H0*** LINEAR AND/OR SPHERICAL COMPONENTS OF PHASE ARE BEI	CYCLE9	85
	ANG REMOVED ***/)	CYCLE9	86
	ITMAX=30	CYCLE9	87
	SPHTOL=.001	CYCLE9	88
	ICKA=0	CYCLE9	89
	ICKH=0	CYCLE9	90
	PI=3.141592	CYCLE9	91
	DELX = X(2)-X(1)	CYCLE9	92
	AATOT=0.0	CYCLE9	93
	AYTOT=0.0	CYCLE9	94
	KOUNT = 0	CYCLE9	95
	EAX = 0.0	CYCLE9	96
	EAY = 0.0	CYCLE9	97
	ENP = 0.0	CYCLE9	98
	HAUCUM = 1.E50	CYCLE9	99
	RUOLD = 1.E50	CYCLE9	100
	AAOLD = AX	CYCLE9	101
	AYOLD = AY	CYCLE9	102
	POW = 0.0	CYCLE9	103
	DO 20 J=1,NPY	CYCLE9	104
	DO 20 I=1,NPTS	CYCLE9	105
	IJ = 1 + (J-1)*NPTS	CYCLE9	106
	CFIL(I,J) = CU(IJ)	CYCLE9	107
	POW = CUM(IJ*2-1)**2 + CUM(IJ*2)**2 + POW	CYCLE9	108

C	POW = CFIL(I,J)*CONJG(CFIL(I,J))*POW	CYCLE9	109
20	CONTINUE	CYCLE9	110
	POW = POW*DELX**2	CYCLE9	111
	NLIMX = NPYS-1	CYCLE9	112
	NLIMY = NPY-1	CYCLE9	113
	IF (NPYS.NE.NPY)NLIMY=NPY	CYCLE9	114
	WRITE(6,180)	CYCLE9	115
180	FORMAT (27X,33MINTERMEDIATE OPTIMIZATION RESULTS //	CYCLE9	116
	A 1UM ITEMATION,5X,5MFUCAL,7X,6MHAUCUR,9X,3MATX,10X,3MATY,	CYCLE9	117
	8 8X,5MATOT,8X,5MATIOT)	CYCLE9	118
25	IF (IPS .EQ. 2) GO TO 54	CYCLE9	119
	KOUNT = KOUNT+1	CYCLE9	120
	CSUMX = (0.0,0.0)	CYCLE9	121
	CSUMY = (0.0,0.0)	CYCLE9	122
	DO 30 J=2,NLIMY	CYCLE9	123
	J1=J+1	CYCLE9	124
	JM=J-1	CYCLE9	125
	IF (J.EQ.NPY)J1=J	CYCLE9	126
	CM=CFIL(I,J)	CYCLE9	127
	CA=CFIL(I2,J)	CYCLE9	128
	CU 30 I=2,NLIMX	CYCLE9	129
	CAA=CFIL(I,J1)	CYCLE9	130
	CCC=CFIL(I,JM)	CYCLE9	131
	CC=CB	CYCLE9	132
	CB=CA	CYCLE9	133
	CA=CFIL(I+1,J)	CYCLE9	134
	CCNJ = CONJG(CM)	CYCLE9	135
	CSUMX = CCNJ*(CA-CC)/2.0*CSUMX	CYCLE9	136
	CSUMY = CCNJ*(CAA-CCC)/2.0*CSUMY	CYCLE9	137
30	CONTINUE	CYCLE9	138
	CAX = CSUMX*DELX	CYCLE9	139
	CAY = CSUMY*DELY	CYCLE9	140
	ATLX =AIMAG(CAX)/POW	CYCLE9	141
	ATLY =AIMAG(CAY)/POW	CYCLE9	142
	IF (NPYS.EQ.NPY) GO TO 52	CYCLE9	143
	ATLY=0.0	CYCLE9	144
52	ATX=-ATLX*WL/(2.*PI)	CYCLE9	145
	ATY=-ATLY*WL/(2.*PI)	CYCLE9	146
	AXTOT=AXIOT+ATX	CYCLE9	147
	AYTOT=AYIOT+ATY	CYCLE9	148
	AX=AX+ATX	CYCLE9	149
	AY=AY+ATY	CYCLE9	150
	DO 40 J=1,NPY	CYCLE9	151
	J1=(J-1)*NPYS	CYCLE9	152
	ATLYY = ATLY * X(J)	CYCLE9	153
	DO 40 I=1,NPYS	CYCLE9	154
	INDX=J1+I	CYCLE9	155
	PHI = ATLX*A(I) * ATLYY	CYCLE9	156
	CFACT = CMPLX (COS(PHI),SIN(PHI))	CYCLE9	157
C	CFACT = CEXP(CMPLX(0.,ATLX*A(I)+ATLYY*X(J)))	CYCLE9	158
	CU(INDX)=CU(INOX)/CFACT	CYCLE9	159
	CFIL(I,J)=CFIL(I,J)/CFACT	CYCLE9	160
40	CONTINUE	CYCLE9	161
	EAX = 0.0	CYCLE9	162
	EAY = 0.0	CYCLE9	163
	IF (ABS(AA) .GT. 0.0)EAX=ABS(1.0-AXOLD/AX)	CYCLE9	164
	IF (ABS(AA) .GT. 0.0)EAY=ABS(1.0-AYOLD/AY)	CYCLE9	165
	ICR = 1	CYCLE9	166
	IF (EAX.LT.0.05.AND.EAY.LT.0.05)ICR=0	CYCLE9	167
	AXOLD = AX	CYCLE9	168
	AYOLD = AY	CYCLE9	169
	IF (IPS .EQ. 1) GO TO 70	CYCLE9	170
C	*****	CYCLE9	171
C	* THE FOLLOWING CALCULATIONS DETERMINE THE LEAST SQUARES SPHERICAL*	CYCLE9	172
C	* FIT TO THE PHASE GRADIENT,---THE RESULT IS 2*PI / (WL*R). *	CYCLE9	173
C	* R = THE RADIUS OF CURVATURE OF THE PHASE FRONT. *	CYCLE9	174
C	***** FURHAM 11/25/74 *****	CYCLE9	175
54	T= 0.0	CYCLE9	176
	DO 55 J = 1, NPY	CYCLE9	177
	DO 55 I = 1, NPYS	CYCLE9	178
	FMAG = CFILR(2*I-1,J)**2 * CFILR(2*I,J)**2	CYCLE9	179

C	FMAG = CFIL(I,J) * CONJG(CFIL(I,J))	CYCLE9	180
	T = FMAG * (X(I)**2 + X(J)**2) * T	CYCLE9	181
55	CONTINUE	CYCLE9	182
	TINT = T * UELX * UELX	CYCLE9	183
	CHA = (0.,0.)	CYCLE9	184
	CHY = (0.,0.)	CYCLE9	185
	DO 60 J = 2 ,NLIMY	CYCLE9	186
	J1=J+1	CYCLE9	187
	JM=J-1	CYCLE9	188
	IF(J.EQ.NPY)J1=J	CYCLE9	189
	CB=CFIL(1,J)	CYCLE9	190
	CA=CFIL(2,J)	CYCLE9	191
	DO 60 I=2,NLIMX	CYCLE9	192
	CAA=CFIL(1,J1)	CYCLE9	193
	CCC=CFIL(1,JM)	CYCLE9	194
	CC=CB	CYCLE9	195
	CB=CA	CYCLE9	196
	CA=CFIL(1+1,J)	CYCLE9	197
	CCNJ = CONJG(CB)	CYCLE9	198
	CHA = CCNJ*(CA-CC)*X(I)/2.0*CHA	CYCLE9	199
	CHY = CCNJ*(CAA-CCC)*X(J)/2.0*CHY	CYCLE9	200
60	CONTINUE	CYCLE9	201
	CHINT = UELX * (CHY * CHA)	CYCLE9	202
	B = -AIMAG(CHINT) / TINT	CYCLE9	203
	IF(ABS(B).GT.(2.*PI/WL/1.E50)) FOCAL = 2*PI/(WL*B)	CYCLE9	204
	HAUCUR = (FOCAL*HAUCUR)/(FOCAL+HAUCUR)	CYCLE9	205
	IF(ABS(HAUCUR).GT.0.0)END=ABS(1.0-HDULD/HAUCUR)	CYCLE9	206
	ICKR=1	CYCLE9	207
	HDULD = HAUCUR	CYCLE9	208
	IF(END.LE.SPHTOL)ICKR=0	CYCLE9	209
C	CHAD = CMPLX(0.0,PI/(WL*FOCAL))	CYCLE9	210
	PIOWLF = PI/(WL*FOCAL)	CYCLE9	211
	DO 80 J=1,NPY	CYCLE9	212
	YSU = X(J)**2	CYCLE9	213
	DO 80 I=1,NPTS	CYCLE9	214
	I2 = 2*I	CYCLE9	215
	I2M1 = I2 - 1	CYCLE9	216
	PHI = (X(I)**2 + YSU) * PIOWLF	CYCLE9	217
	SINP = SIN(PHI)	CYCLE9	218
	COSP = COS(PHI)	CYCLE9	219
	CURS = CFILX(I2M1,J)	CYCLE9	220
	CFILX(I2M1,J) = CURS*COSP - CFILX(I2,J)*SINP	CYCLE9	221
80	CFILX(I2,J) = CURS*SINP + CFILX(I2,J)*COSP	CYCLE9	222
C	80 CFIL(I,J)=CFIL(1,J)*CEXP((X(I)**2+X(J)**2)*CHAD)	CYCLE9	223
70	UMAX = 0.0	CYCLE9	224
	WRITE(6,190) KOUNT,FOCAL,HAUCUR,ATX,ATY,AXTOT,AYTOT	CYCLE9	225
190	FORMAT(1X,15.4X,6G13.4)	CYCLE9	226
	IF(FOCAL.GT.-4.E05.AND.FOCAL.LT.6.E05.AND.KOUNT.LT.ITMAX)GO TO 25	CYCLE9	227
	IF((ICKR.GT.0.OR.ICKR.GT.0).AND.KOUNT.LT.ITMAX) GO TO 25	CYCLE9	228
	IF(IPS.EQ.1.OR.IPS.EQ.3)WRITE(6,201)AXTOT,AYTOT	CYCLE9	229
201	FORMAT(/20X,16HLINER COMPONENT//	CYCLE9	230
	X 10X,8MTILT IN 1MX9M = A(X) =.612.4.8M RADIANS/	CYCLE9	231
	X 10X,8MTILT IN 1MY9M = A(Y) =.612.4.8M RADIANS)	CYCLE9	232
	IF(IPS.GE.2)WRITE(6,67)HAUCUR	CYCLE9	233
67	FORMAT(/20X,19MSPHERICAL COMPONENT//	CYCLE9	234
	X 10X,32MPHASE FRONT CURVATURE = HAUCUR =.612.4.3M CM//)	CYCLE9	235
	RETURN	CYCLE9	236
	END	CYCLE9	237

37. SUBROUTINE ERF

- a. Purpose -- The function ERF generates the error function

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (267)$$

or its complement, $1-\text{erf}(x)$, for any input value of x . This subroutine is a copy of the ERF function available from the AFWL scientific program library. Figure 73 shows the Subroutine ERF flow chart.

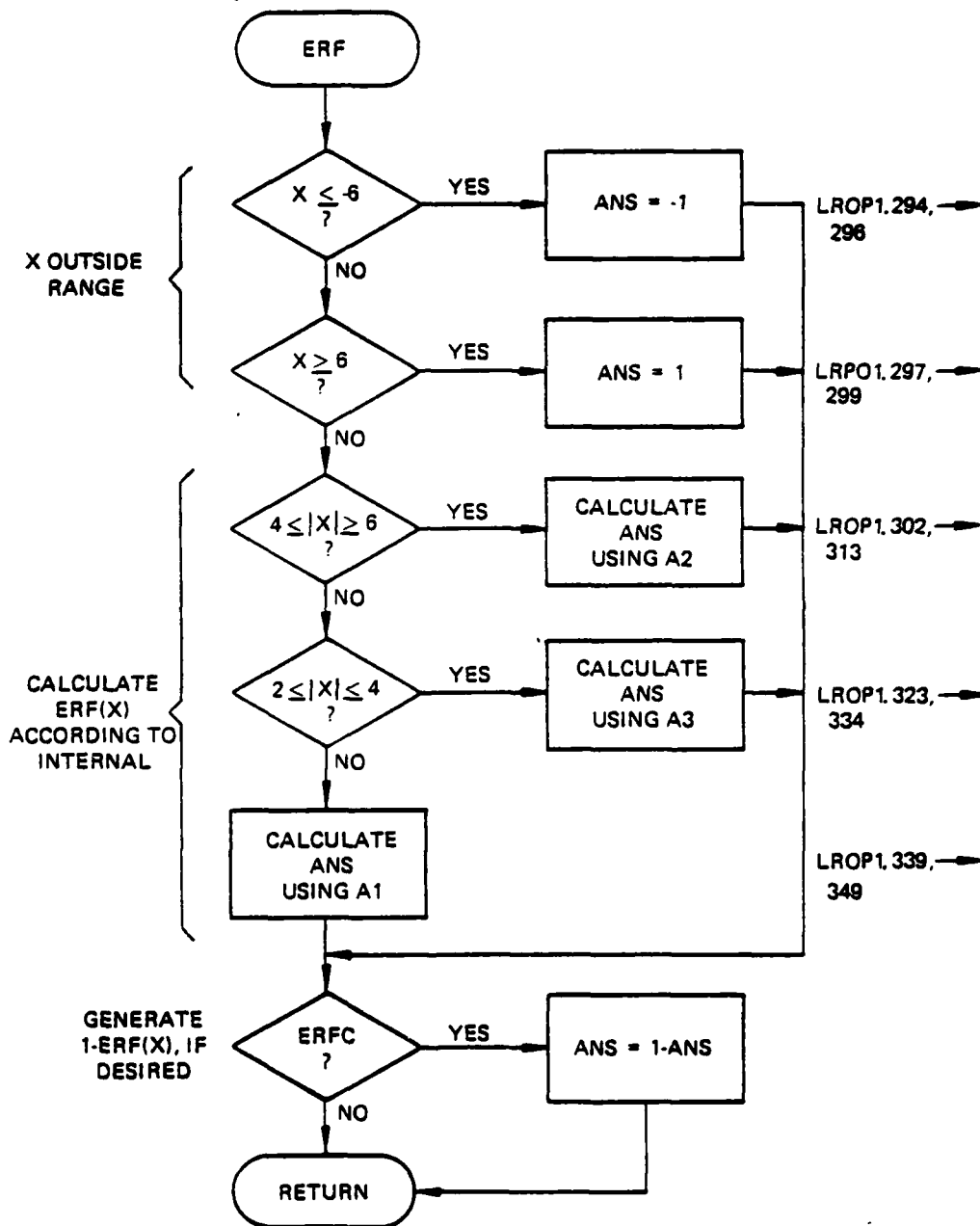


Figure 73. Subroutine ERF flow chart.

b. Relevant formalism -- The error function integral is approximated over discrete intervals of the argument, x, by Tchebichef (Chebychev) polynomials. These polynomials are evaluated in a loop which combines the recurrence relations for generating the polynomials and a running summation of the terms as they are generated. Coefficients for the polynomials are provided in a data statement for three discrete ranges of the argument. Argument values outside this range will return a zero (0).

Argument List

ANS error function value returned to calling program
 KODE flag to indicate computation of erf(x) or 1-erf(x)
 XX error function argument

Relevant Variables

A1 array of coefficients used in the polynomial expansion over the range $|xx| \leq 2$.
 A2 coefficient array for the range $4. \leq |xx| \leq 6$.
 A3 coefficient array for the range $2. < |xx| < 4$.

SUBROUTINE ERF 76/176 OPT=1 FIN 4.6+452 04/27/79 12.23.47

```

SUBROUTINE ERF(KODE,XX,ANS)
  COMPUTES BY CHEBYSHEV EXPANSIONS ON INTERVALS.
  KODE=1 COMPUTES ERF(X)
  KODE=2 COMPUTES ERFC(X)=1-ERF(X)
  CDC 6600 ROUTINE
  1-4-72
  DIMENSION A1(31),A2(27),A3(16)
  DATA(A1(1),I=1,31)/2.96622112816961E+0,0.,-6.02142146773189E-1,0.,
  11.37989661379662E-1,0.,-2.78325425294437E-2,0.,4.84159904486783E-3
  2,0.,-7.31727937169453E-4,0.,4.72419888637174E-5,0.,-1.149851311618
  304E-5,0.,1.22264871646433E-6,0.,-1.17982030973170E-7,0.,1.04140177
  4691278E-8,0.,-8.4659532945225E-10,0.,6.37620443498960E-11,0.,-4.4
  57177281962215E-12,0.,2.93540222982101E-13,0.,-1.83283038964141E-14
  6/
  DATA(A2(1),I=1,27)/1.970705272257540.,-1.43397402717750E-2,0.,
  12.47361692202619E-4,0.,-7.80351604336237E-6,0.,4.3313342034728E-7,
  20.,-2.362150026241E-8,0.,1.51549676581E-9,0.,-1.1084939856E-10,
  30.,9.04259014E-12,0.,-8.0947054E-13,0.,7.853856E-14,0.,
  4-8.17918E-15,0.,9.0715E-16,0.,-1.0646E-16/
  DATA(A3(1),I=1,16)/1.06663088531943E+0,1.78876062094436E-2,-3.8017
  1529380401E-3,6.97111435023601E-4,-1.16388846083892E-4,1.813676759
  232619E-5,-2.67719439785138E-6,3.77701329909996E-7,-5.1249114250148
  32E-8,6.71870395763107E-9,-8.54019648112644E-10,1.05544302186899E-1
  40,-1.27108499000124E-11,1.49441348185064E-12,-1.71382907865335E-13
  5,2.08849564313469E-14/
  LNUP1 258
  LNUP1 259
  LNUP1 260
  LNUP1 261
  LNUP1 262
  LNUP1 263
  LNUP1 264
  LNUP1 265
  LNUP1 266
  LNUP1 267
  LNUP1 268
  LNUP1 269
  LNUP1 270
  LNUP1 271
  LNUP1 272
  LNUP1 273
  LNUP1 274
  LNUP1 275
  LNUP1 276
  LNUP1 277
  LNUP1 278
  LNUP1 279
  LNUP1 280
  LNUP1 281
  LNUP1 282
  LNUP1 283
  LNUP1 284
  LNUP1 285
  LNUP1 286

```

C	DATA WTP1,XLIM/1.7724538509052, 2.58408528684382E+1/ DATA N1,N1M1,N2,N2M1,N3,N3M1/31,30,27,26,16,15/	LHOP1 287 LHOP1 288 LHOP1 289 LHOP1 290 LHOP1 291 LHOP1 292 LHOP1 293 LHOP1 294 LHOP1 295 LHOP1 296 LHOP1 297 LHOP1 298 LHOP1 299 LHOP1 300 LHOP1 301 LHOP1 302 LHOP1 303 LHOP1 304 LHOP1 305 LHOP1 306 LHOP1 307 LHOP1 308 LHOP1 309 LHOP1 310 LHOP1 311 LHOP1 312 LHOP1 313 LHOP1 314 LHOP1 315 LHOP1 316 LHOP1 317 LHOP1 318 LHOP1 319 LHOP1 320 LHOP1 321 LHOP1 322 LHOP1 323 LHOP1 324 LHOP1 325 LHOP1 326 LHOP1 327 LHOP1 328 LHOP1 329 LHOP1 330 LHOP1 331 LHOP1 332 LHOP1 333 LHOP1 334 LHOP1 335 LHOP1 336 LHOP1 337 LHOP1 338 LHOP1 339 LHOP1 340 LHOP1 341 LHOP1 342 LHOP1 343 LHOP1 344 LHOP1 345 LHOP1 346 LHOP1 347 LHOP1 348 LHOP1 349 LHOP1 350 LHOP1 351 LHOP1 352 LHOP1 353 LHOP1 354 LHOP1 355 LHOP1 356 LHOP1 357 LHOP1 358
C	X=XX GO TO (100,200),KQUE 100 CONTINUE IF(X.LE.-6.) 10,20 10 ANS=-1. RETURN 20 IF(X.LT.6.) GO TO 12 ANS=1. RETURN 12 IF(X.LT.+.) GO TO 30 ASSIGN 26 TO ISET 61 CONTINUE Z=X/X & TZ=Z*Z B2=0. B1=0. DO 25 I=1,N2M1 J=N2-I+1 TEMP=B1 B1=TZ*B1-B2*A2(J) B2=TEMP 25 CONTINUE ANS=Z*B1-B2*A2(1)/2. ANS=(EXP(-X*X)/(X*WTP1))*ANS GO TO ISET,(26,27,28,29) 26 ANS=1.-ANS 27 RETURN 29 ANS = -1.*ANS RETURN 30 CONTINUE IF(X.GT.2.) 31,40 31 CONTINUE ASSIGN 26 TO ISET 35 CONTINUE Z=X-3. & TZ = Z*Z B2=0. B1=0. DO 36 J=1,N3M1 K=N3-J+1 TEMP=B1 B1=TZ*B1-B2*A3(K) B2=TEMP 36 CONTINUE ANS=Z*B1-B2*A3(1)/2. ANS=EXP(-X*X)*ANS/X GO TO ISET,(26,27,28,29) 40 CONTINUE IF(X.LT.-2.) GO TO 50 ASSIGN 27 TO ISET 42 CONTINUE Z=X/2. & TZ=Z*Z B2=0. B1=0. DO 45 I=1,N1M1 J=N1-I+1 TEMP=B1 B1=TZ*B1-B2*A1(J) B2=TEMP 45 CONTINUE ANS=(X/2.)*(Z*B1-B2*A1(1)/2.) GO TO ISET,(26,27,28,29) 50 CONTINUE IF(X.GT.-4.) 51,60 51 CONTINUE ASSIGN 29 TO ISET & X=X & GO TO 35 60 CONTINUE X=-XX ASSIGN 29 TO ISET GO TO 61	

```

200 CONTINUE
    IF(X.GT.-6.) GO TO 205
    ANS=2. 5 RETURN
C
205 IF(X.LT.XLIM) GO TO 210
    ANS=0. 5 RETURN
210 CONTINUE
    IF(X.LT.+4) GO TO 215
    ASSIGN 27 TO ISET
    GO TO 61
C
215 IF(X.GT.2.) GO TO 220
    IF(X.LT.-2.) GO TO 225
    ASSIGN 26 TO ISET
    GO TO 42
C
220 ASSIGN 27 TO ISET
    GO TO 35
C
225 IF(X.GT.-4.) GO TO 230
    ASSIGN 28 TO ISET
    X=-X 5 GO TO 61
    28 ANS=2.-ANS 5 RETURN
C
230 ASSIGN 28 TO ISET 5 X=-X
    GO TO 35
    END

```

```

LNUP1 359
LNUP1 360
LNUP1 361
LNUP1 362
LNUP1 363
LNUP1 364
LNUP1 365
LNUP1 366
LNUP1 367
LNUP1 368
LNUP1 369
LNUP1 370
LNUP1 371
LNUP1 372
LNUP1 373
LNUP1 374
LNUP1 375
LNUP1 376
LNUP1 377
LNUP1 378
LNUP1 379
LNUP1 380
LNUP1 381
LNUP1 382
LNUP1 383
LNUP1 384
LNUP1 385

```


SECTION IV
USER FAMILIARIZATION PACKAGE

The following section contains sample input to run the SOQ code and to logically define the sequence of input to model a sample resonator or optical train the following examples are included:

1. Propagate for Users Guide - Camp
2. Propagate for Users Guide - Vamp
3. Quality for Users Guide
4. Design of a Bare Confocal Resonator
5. Resonator for Users Guide - Bare
6. Resonator for Users Guide - Loaded
7. Sample Code Update

1. PROPAGATE FOR USERS GUIDE - CAMP

```

JRAPC,SIMFX,P4000,T177,EC1. PROPAGATE FOR USERSGUIDE - CAMP
ACCOUNT(JRALT,*****-***.LNO,1731)
GETPF(OLDPL,SOQ77128,IO=***** )
UPDATE(F,W)
FTN(I,LCM=1,PL=20000,L=0,A)
RETURN(OLDPL)
COPYCH,INPUT,TAPES.
RE=IND,TAPES.
HFLFC(430)
LGO(PL=60000)
HFLFC(1)
*EUR
      PROPAGATE - CAMP
$START WNL=0.00106, NCALL=2, DCAL=15., NNPTS=128,
      IN=8, NDRX=0.0, DRY=0.0, AMPGES=20.0, NGAUSS=0.0,
      RESTRT=.FALSE., PLOTS=1.0, IN=5,
      SYMTRC=.FALSE., PHIRAD=0.0, SEND)
      PROPAGATE - CAMP
$CONTRL IFLOW=4. SEND
      APERTURE THE PLANE WAVE TO 10. CM.
$APTUR DOUT=10., DIN=0., $END
$CONTRL IFLOW=8. SEND
      PLOT THE INITIAL PLANE WAVE
$PLOT $END
      INITIAL PLANE WAVE
$CONTRL IFLOW=3. SEND
      PROPAGATE THE FIELD 4000 CM. USING CONSTANT AREA MESH
$PROPGT DFLZ=4000., RDCURV=0., WINDOK=0.1, WINDOK=0.1,

```

```

      IIFG=1.    IITH=0.    IIPS=0.    SEND
$CONTRL IFLOW=8.  SEND
      PLOT PROPAGATED FIELD
$PLOT $END
      PROPAGATED FIELD
$CONTRL IFLOW=9.  SEND
      RETURN TO MAIN
$START WWL=-1..  SEND
*EOR
*EOF

```

2. PROPAGATE FOR USERS GUIDE - VAMP

```

100=JRAUG,STMFY,P60,T77,EC1. PROPAGATEFORUSERSGUIDEVAMP, ID=LREPPEF
110=ACCOUNT(JRALY,00011498-1EL,LRO,1487)
120=ATTACH(OLDPL,SOQ77128, ID=LROPJRA, ST=ANY)
130=UPDATE(F)
140=FTN(I,LCM=1,PL=20000,L=0)
150=RETURN(OLDPL)
160=COPY, INPUT, TAPES.
170=REWIND, TAPES.
180=RFLEC(430)
190=LGO(PL=60000)
200=RFLEC(1)
210=*EOR
220=*EOR
230= PROPAGATE A MIRRORRED PLANE WAVE A DISTANCE DELZ - VAMP
240= $START WWL=0.00106, NCALL=2, DCAL=5.6, NNPTS=128,
250= IB=8, DDRX=0.0, DDRY=0.0, AMPGES=20.0, DGAUSS=0.0,
260= RESTRY=.FALSE., PLOTS=1.0, IN=5,
270= SYMTRC=.FALSE., PHIRAD=0.0, SEND
280= PROPAGATE A MIRRORRED PLANE WAVE A DISTANCE DELZ - VAMP
290= $CONTRL IFLOW=2, SEND
300= APPLY A MIRROR TO THE PLANE WAVE
310= $MIRROR DIAOUT=4.0, DIAIN=0.0, XMPOS=0.0, YMPOS= 0.0,
320= RADC=-400., RMIR=1., =SEND
330= $CONTRL IFLOW=8, SEND
340= PLOT THE MIRRORRED PLANE WAVE FIELD
350= $PLOT SEND
360= INITIAL MIRRORRED PLANE WAVE FIELD
370= $CONTRL IFLOW=3, SEND
380= PROPAGATE THE FIELD 200. CM. USING VARIABLE AREA MESH
390= $PROPGT DELZ=200., RDCURV=0., WINDOX=0.1, WINDOK=0.1,
400= IIFG=2, IITH=1, IIPS=0, SEND
410= $CONTRL IFLOW=8, SEND
420= PLOT PROPAGATED FIELD
430= $PLOT SEND
440= PROPAGATED FIELD
450= $CONTRL IFLOW=9, SEND
460= RETURN TO MAIN PROGRAM
470= $START WWL=-1.. SEND
480=*EOR
490=*EOF

```

3. QUALITY FOR USERS GUIDE

```

JXADD,SIMPX,P4000,1177,EC1.    QUALITY FOR USERS GUIDE
ACCOUNT(JRALT,*****-***,L40,173)
GETPF(OLDPL,50077128,10=*****)
GETPF(TAPEA,USERSGUIDEHARECU,10=*****)
UPDATE(F,W)
FTN(1,LCM=1,PL=20000,L=0,A)
RETURN(OLDPL)
COPYCR.INPUT.TAPF5.
REWIND.TAPF5.
WFLEC(430)
LGO(PL=60000)
WFLEC(1)
*FOR
  FIND THE QUALITY OF THE FIELD
  $START WWL=0.00106, NCALL=2, DCAL=13.78, NNPTS=128,
  IH=8, DDRX=0.0, DDY=0.0, AMPGFS=1.0, NGAUSS=0.0,
  RESTR= .TRUE., PLOTS=1.0, IN=5,
  SYMTRC=.FALSE., PHIRAD=0.0, $FND
  FIND THE QUALITY OF THE FIELD
  $CONTROL IFLOW=8, $END
  PLOT THE FIELD
  $PLOT $END
  FIELD AT INPUT
  $CONTROL IFLOW=9, $END
  RETURN TO MAIN PROGRAM FOR QUALITY CALCULATION
  $START NCALL=3, $END
  DB = 10.64
  $PLOT DB=10.64, ISAV=0, IULT=0, IPHASE=3, $FND
  DB = 10.64
  $START WWL=-1.0, $FND
*FOR
*FOR

```

4. DESIGN A BARE CONFOCAL RESONATOR

Assume that one wishes to design a positive branch, unstable bare resonator with a collimated output beam for a given geometric coupling C_g , length L , and concave mirror size (a_1). To solve this problem design a confocal resonator in the following fashion:

Geometric Resonator Design (Fig. 74).

Define the following parameters

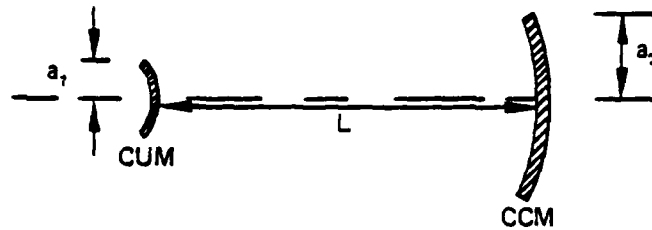


Figure 74. Geometric resonator design.

Recall the definition of geometric coupling.

$$C_g = \frac{A_{OUT}}{A_{TOTAL}} = \frac{\pi a_2^2 - \pi a_1^2}{\pi a_2^2} = 1 - \frac{1}{\left(\frac{a_2}{a_1}\right)^2} \quad (268)$$

But $M = a_2/a_1$ is the magnification of the resonator, thus

$$C_g = C_g = 1 - \frac{1}{M^2} \quad (269)$$

Or inverting this expression, one finds

$$M = \frac{1}{\sqrt{1-C_g}} \quad (270)$$

Given the magnification and length of the resonator, one can find the required mirror radii of curvature, since for an aligned confocal resonator both the convex and concave mirror foci are coincident. Figure 75 describes this coincident feature.

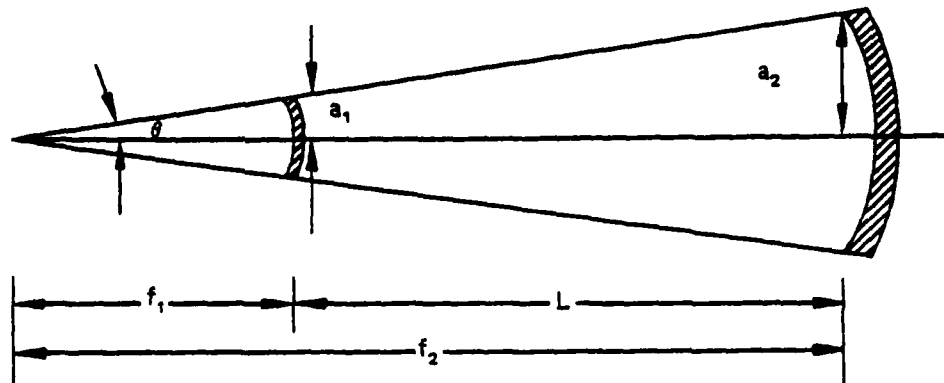


Figure 75. Required mirror radii of curvature.

a. The focal lengths can be related to the magnification by noting that

$$\tan \theta = \frac{a_1}{f_1} = \frac{a_2}{f_2} = \frac{a_3}{f_3} \quad (271)$$

therefore

$$M = \frac{a_2}{a_1} = \frac{f_2}{f_1} = \frac{f_1 + L}{f_1} \quad (272)$$

The focal lengths are then found to be

$$f_1 = \frac{L}{M-1} \quad \text{and} \quad f_2 = Mf_1 = \frac{ML}{M-1} \quad (273)$$

Since the radius of curvature of a mirror is twice its focal length, the two radii of curvature are

$$R_1 = \frac{-2L}{M-1} \quad R_2 = -MR_1 = \frac{2ML}{M-1} \quad (274)$$

where the negative sign indicates a convex mirror and the positive, a concave. For example, if $L = 200$ cm and $C_g = 0.75$, the magnification and radii are found to be

$$M = \frac{1}{\sqrt{0.25}} = 2 \quad (275)$$

$$R_1 = \frac{-(2)(200)}{(1)} = -400 \text{ cm} \quad \text{and} \quad R_2 = -(2)(-400) = 800 \text{ cm} \quad (276)$$

b. Tube Fresnel number -- The tube Fresnel number for this resonator can be found by the fact that the expanding pass propagation distance L has an equivalent collimated propagation length of ML so the round trip collimated propagation distance is $(M + 1)L$. The tube Fresnel number is then (assuming the CVM is 2.0 cm in radius and the beam has a wave length of 10.6 μm).

$$N_T = \frac{a_1}{(M+1)L\lambda} = \frac{(2)^2}{(5)(200)(10.6 \times 10^{-4})} = 6.29 \quad (277)$$

c. Computer requirements

(1) Overlap -- Since the beam diffracts during propagation, it is necessary to have a large enough calculation region to always contain the beam. The required overlap can be calculated according to Sziklas and Siegman (Ref. 2) as

$$G \geq 1 + \frac{1}{2\pi^2 N_T \epsilon} \quad (278)$$

where ϵ is the tolerance on fractional energy loss during propagation. Taking this to be 0.02, one finds the guardband to be

$$G \geq 1 + \frac{1}{2\pi^2 (6.29)(0.02)} = 1.4 \quad (279)$$

Thus the initial calculation region must be at least G times the beam size:

$$DCALC = 1.4 \times 2 \times 2 = 5.6 \text{ cm} \quad (280)$$

(2) Number of points required -- Sziklas and Siegman also show that in order to adequately sample the beam, the number of points in each dimension must obey the following inequality:

$$N_p \geq 4G(G+1)N_T \quad (281)$$

This becomes

$$N_p \geq 4(1.4)(2.4)(6.29) = 85 \quad (282)$$

Standard input for the SOQ deck is 128 by 128 so this criterion is satisfied.

d. SOQ input -- As a result of the above discussion, the parameters used for a bare resonator test case could be the following:

NPTS = 128
 DCAL = 5.6 cm
 CVM: RADC = -400 cm
 DIAOUT = 4.0 cm
 DIAIN = 0.0 cm
 DELZ = 200.0 cm
 CCM: RADC = 800.0 cm
 DIAOUT = 8.0 cm
 DIAIN = 0.0 cm

5. RESONATOR FOR USERS GUIDE - BARE

```

JRAHR,STMFx,P4000,T177,EC1. RESONATORFORUSERSGUIDEBARE
ACCOUNT(JRALT,*****-***,LWU,1731)
REQUEST(TAPER,*PF)
REQUEST(TAPE9,*PF)
GETPF(OLDPL,50077128, ID=*****)
UPUATE(F,N,w,L=0)
FTN(I,LCM=I,PL=20000,L=0,A)
RETURN(OLDPL)
COPYCR,INPUT,TAPES.
REWIND,TAPES.
GETPF(PEH,USFRSGUIDERARECUSM, ID=*****)
GETPF(PE9,USFRSGUIDERARECU, ID=*****)
RFLEC(430)
LGO(PL=20000)
RFLEC(1)
PURGE(WAPER,USFRSGUIDERARECUSM, ID=*****,LC=1)
PURGE(WAPE9,USFRSGUIDERARECU, ID=*****,LC=1)
CATALOG(TAPER,USFRSGUIDEBARECUSM, ID=*****,RP=999)
CATALOG(TAPE9,USFRSGUIDEBARECU, ID=*****,RP=999)
*EOR
*EOR
SIMPLF CONFOCAL BARE RESONATOR - M=2, NTUBE=5.03
$START WWL=0.00106, NCALL=2, DCAL=6.4, NNPTS=128,
IB=8, DDRX=0.0, DORY=0.0, AMPGES=20.0, NGAUSS=0.0,
RESTR= .TRUE., PLOTS=1.0, IN=5,
SYMTRC=.FALSE., PHIRAD=0.0, SEND
SIMPLF CONFOCAL BARE RESONATOR - M=2, NTUBE=5.03
$CONTROL IFLOW=2, SEND
APPLY CVM MIRROR
$MIRROR RADC=-2000., DIAOUT=4.0, DIAIN=0.0, RMIR=.997,
DELTA=0.0, ANGXX=0.0, ANGY=0.0, XMPOS=0.0, YMPOS=0.0,
DISTF=0.0, SEND
$CONTROL IFLOW=8, SEND
PLOT THE CVM FIELD
$PLOT SEND
THE CVM FIELD
$CONTROL IFLOW=3, SEND
  
```

```

PROPAGATE THE FIELD TO THE CCM USING VAMP
$PROPGT DELZ=1000., WINDOX=0.1, WINDOK=0.1, IIFG=2, IIPS=0,
  IITH=1, ROCURV=1000., $END
$CONTRL IFLOW=8, $END
PLOT THE FIELD INCIDENT ON CCM
$PLOT $END
FIELD INCIDENT ON CCM
$CONTRL IFLOW=2, $END
APPLY CCM
$MIROR RADIC=4000., DIAOUT=8., $END
$CONTRL IFLOW=8, $END
PLOT THE CCM FIELD
$PLOT $END
FIELD AFTER ON CCM
$CONTRL IFLOW=3, $END
PROPAGATE THE FIELD BACK TO THE CVM USING CONSTANT AREA MESH
$PROPGT DELZ=1000., WINDOX=0.1, WINDOK=0.1, IIFG=1, IIPS=0,
  IITH=0, ROCURV=0.0, $END
$CONTRL IFLOW=6, $END
FIELD CUTOUT AND INTERPOLATION FOR THE NEXT PASS
$CUTOUT DREAM=4.0, OVHLAP=1.6, DXXR=0., DYYR=0., MAXIT=3,
  AVCUSM=0.0, $END
$CONTRL IFLOW=8, $END
PLOT THE FIELD INCIDENT ON CVM
$PLOT $END
FIELD INCIDENT ON CVM
$CONTRL IFLOW=7, $END
CONVERGENCE TEST
$CONTRL IFLOW=9, $END
RETURN TO MAIN PROGRAM
$START WWL=-1., $END
*EUR
*EUF

```

6. RESONATOR FOR USERS GUIDE - LOADED

```

JWALR,STMFx,P4000,T177,EC1, RESONATORFORUSERSGUIDELOADED
ACCOUNT(JRALT,*****-***,LRU,1731)
REQUEST(TAPE8,*PF)
REQUEST(TAPE9,*PF)
REQUEST(TAPE11,*PF)
REQUEST(TAPE12,*PF)
REQUEST(TAPE13,*PF)
GETPF(OLDPL,50Q77128, ID=***** )
UPDATE(F,N,w,L=0)
FTN(I,LCM=1,PL=20000,L=0,A)
RETURN(OLDPL)
COPYCH,INPUT,TAPES.
NEWIND,TAPES.
GETPF(TAPE8,USERSGUIDELOADEDUCUSM, ID=***** )
GETPF(TAPE9,USERSGUIDELOADEDUCU, ID=***** )
GETPF(TAPE11,USERSGUIDELOADEDG11, ID=***** )
GETPF(TAPE12,USERSGUIDELOADEDG12, ID=***** )
GETPF(TAPE13,USERSGUIDELOADEDG13, ID=***** )
GETPF(TAPE31,OPD1131141PTBSECCONTXY, ID=***** )
RFLEC(430)
LGO(PL=60000)
RFLEC(1)

```



```

PURGE (WAPE8,USERSGUIDELOADEDUCUM, ID=*****.LC=1)
PURGE (WAPE9,USERSGUIDELOADEDUCU, ID=*****.LC=1)
PURGE (WAPE11,USERSGUIDELOADEDUCG11, ID=*****.LC=1)
PURGE (WAPE12,USERSGUIDELOADEDUCG12, ID=*****.LC=1)
PURGE (WAPE13,USERSGUIDELOADEDUCG13, ID=*****.LC=1)
CATALOG (TAPE8,USERSGUIDELOADEDUCUM, ID=*****.RP=999)
CATALOG (TAPE9,USERSGUIDELOADEDUCU, ID=*****.RP=999)
CATALOG (TAPE11,USERSGUIDELOADEDUCG11, ID=*****.RP=999)
CATALOG (TAPE12,USERSGUIDELOADEDUCG12, ID=*****.RP=999)
CATALOG (TAPE13,USERSGUIDELOADEDUCG13, ID=*****.RP=999)

```

*EUR

*EOR

```

SIMPLE CONFOCAL LOADED RESONATOR - M=2, NTURE=5.03
$START WWL=0.00106, NCALL=2, DCAL=6.4, NNPTS=128,
IB=8, DNRX=0.0, DNRV=0.0, AMPGES=20.0, DGAUSS=0.0,
RESTR= .TRUE., PLOTS=1.0, IN=5,
SYMTIC=.FALSE., PHIRAD=0.0, $END

```

SIMPLE CONFOCAL LOADED RESONATOR - M=2, NTURE=5.03

\$CONTROL IFLOW=2, \$END

APPLY CVM MIRROR

```

$MIRROR RADC=-2000., DIAOUT=4.0, DIAIN=0.0, RMIR=.997,
DELTA=0.0, ANGXX=0.0, ANGY=0.0, XMPOS=0.0, YMPOS=0.0,
DISTF=2.E-7, $END

```

\$CONTROL IFLOW=8, \$END

PLOT THE CVM FIELD

\$PLOT \$END

THE CVM FIELD

\$CONTROL IFLOW=3, \$END

PROPAGATE THE FIELD TO THE CAVITY USING VAMP

```

$PROPGT DEL7=100., WINDOWX=0.1, WINDOWY=0.1, IIFG=2, IIPS=0,
IITR=0, MDCURV=1000., $END

```

\$CONTROL IFLOW=1, \$END

APPLY GDL CAVITY

```

$CAVITY1 NCAVNO=1, NSTE=4, ILR=1, NPLT=0, ZPROPI=0.,
ZPROPO=150., $END

```

```

$CAVITY2 XLEN=24.32, YLEN=11.4, ZLEN=750., XMCAV=6., YMCAV=0.,
NODX=190, NODY=90, NOSEG=3, FLAG=11., MREST=0.,
NGTYPE=0, NGPLOT=0, IPDEN=0, IUSE=-1,
T1=391.2, T2=395.2, T3=1284, TN2=1333.5,
TS=313., PS=.0422, V=171380., PHRCH=18.,
XN2=.8121, XCO2=.1388, XM2U=.0146, XCO=.0044, XU2=.0241,
AVGAIN=.3, $END

```

USERS GUIDE LOADED RESONATOR

\$CONTROL IFLOW=2, \$END

APPLY CCM

```

$MIRROR RADC=4000., DIAOUT=8., $END

```

\$CONTROL IFLOW=8, \$END

PLOT THE CCM FIELD

\$PLOT \$END

FIELD AFTER CCM

\$CONTROL IFLOW=1, \$END

PROPAGATE THE FIELD BACK THROUGH THE CAVITY USING CONSTANT AREA MESH

```

$CAVITY1 NCAVNO=1, NSTE=1, ILR=-1, NPLT=0, ZPROPI=150.,
ZPROPO=100., $END

```

\$CONTROL IFLOW=6, \$END

FIELD CUTOFF AND INTERPOLATION FOR THE NEXT PASS

```

$CUTOFF DIRFAM=4.0, OVLAP=1.0, DXXR=0., DYYR=0., MAXIT=3,
AVCUSM=-1., $END

```

```

$CONTROL IFLOW=8, $FNO)
  PLUT THE FIELD INCIDENT ON CVM
$PLOT $FNO
  FIELD INCIDENT ON CVM
$CONTROL IFLOW=7, $END
  CONVERGENCE TEST
$CONTROL IFLOW=9, $FNO
  RETURN TO MAIN PROGRAM
$START #WL=-1., $END
*EUR
*EUF

```

7. SAMPLE CODE UPDATE

The following file is included to illustrate the set of updates which would be included to add a subroutine to the existing SOQ group of subroutines. The updates are comprehensive in that they illustrate common modifications and include a namelist and subroutine within the beam quality calculation division of the SOQ code.

```

JHAZK,STMPX,P4000,1117,EC1.  ADD ZERNIKE REMOVAL TO SOQ
ACCOUNT(JRALT,*****-***,LR0,1731)
GETPF(OLDPL,S007712H,I0=*****)
UPDATE(F,W)
FTN(I,LCM=1,PL=20000,L=0,A)
RETURN(OLDPL)
COPYCR.INPUT.TAPES.
REWIND.TAPES.
#FLEC(430)
LGO(PL=60000)
#FLEC(1)
*EOR
*ID ZRNIKE
*I GDL.261
  IZERN = 0
*I GDL.315
  IZERN = 0
*I S0077CY1.165
C      = 23  APPLY UP TO 24 ZERNIKES IN UNITS OF WAVES. HEADS ZERNS
*I GDL.29
  LOGICAL FRINGE
*U GDL.295,S0077CY1.167
C      /16 /17 /18 /19 /20 /21 /22 /23 /
  X.140,170,180,190,200,210,365,230),IFLOW
*U GDL.325,S0077CY1.168
C      /16 /17 /18 /19 /20 /21 /22 /23 /
  X.160,170,180,190,200,210,365,230),IFLOW
*I GDL.327
C.....
C      APPLY ZERNIKE
C.....
  230 IZERN = IZERN + 1
  IF (.NOT.INIT) GO TO 244

```

```

FRINGE = .FALSE.
DO 248 I=1,24
248 P(I) = 0.
DO 249 I=1,35
249 PFRNG(I) = 0.
READ (5,ZERNS)
DO 239 I=1,35
239 IF(PFRNG(I).NE.0.) FRINGE=.T.
IF(.NOT.FRINGE) GO TO 241
WRITE(6,245)
245 FORMAT(/5X,*FRINGE COEFFICIENTS BEING CONVERTED TO SQ ORDER.*/)
P(1) = 0.
P(2) = PFRNG(1)
P(3) = PFRNG(2)
P(4) = PFRNG(3)
P(5) = PFRNG(4)
P(6) = PFRNG(5)
P(7) = PFRNG(6)
P(8) = PFRNG(7)
P(9) = PFRNG(9)
P(10) = PFRNG(10)
P(11) = PFRNG(8)
P(12) = PFRNG(11)
P(13) = PFRNG(12)
P(14) = PFRNG(16)
P(15) = PFRNG(17)
P(16) = PFRNG(13)
P(17) = PFRNG(14)
P(18) = PFRNG(18)
P(19) = PFRNG(19)
P(20) = PFRNG(25)
P(21) = PFRNG(26)
P(22) = PFRNG(15)

P(23) = PFRNG(24)
P(24) = PFRNG(35)
IFRTST = 0
DO 246 K=20,23
246 IF(PFRNG(K).NE.0.) IFRTST = 1
DO 243 K=27,34
243 IF(PFRNG(K).NE.0.) IFRTST = 1
IF(IFRTST.EQ.1) WRITE(6,247)
247 FORMAT(/5X,*WARNING - FRINGE COEFFICIENTS OF ORDER 20 THROUGH 23*,
C * AND 27 THROUGH 34 ARE IGNORED*/)
241 DO 242 I=1,24
242 PZSAVE(I,IZERN) = P(I)
PZSAVE(25,IZERN) = R0
244 CALL ZFRN(PZSAVE(25,IZERN),PZSAVE(I,IZERN))
IGNAL = 1
GO TO 999
*D GDL.27
DIMENSION IPLTS(50),PZSAVE(25,10),P(24),PFRNG(35)
*I GDL.33
DATA P,PFRNG/24*0.,35*0./ , R0 / 5. /
*I GDL.243
C
C   NAMELIST /ZERNS/ R0,P,PFRNG
C
C   R0 = RADIUS OVFR WHICH ZERNIKES ARE VALID.

```

```

C      P = ARRAY ZERNIKE COEFFICIENTS.
C      PFRNG = ARRAY FRINGE ZERNIKE COEFFICIENTS (CONVERTED TO P IN GPL).
* I LROPI.385
SUBROUTINE ZERN(R0,P)
LEVEL 2.CUR
COMMON /MELT/ CUR(32768),CFIL(16512),X(128),WL,NPTS,NPY,DRX,DRY
COMPLEX CFIL
DIMENSION P(24)
IF(R0.FQ.0.) GO TO 70
DO 100 IY=1,NPY
  JI = (IY-1)*NPTS
  YSQ = X(IY)**2
  DO 100 IX=1,NPTS
    XSQ = X(IX)**2
    INDX = IX + JI
    R = SQRT(XSQ+YSQ)
52 THET = ATAN2(X(IY),X(IX))
    R = AMINI(R/R0,1.)
    CT = COS(THET)
    C2T = COS(2.*THET)
    C3T = COS(3.*THET)
    C4T = COS(4.*THET)
    CST = COS(5.*THET)

```

JMA/R.51MFX,PL=0001,177.FCI. ADD ZERNIKE REMOVAL TO SOU
 ACCOUNT JRALF.000000000000.LRN.1741
 GETPF(0,0,01,5007712A,1000000000)
 UPDATE(F,W)
 FIN(I,LCM=1,PL=20000,L=0,W)
 RETURN(OLNPL)
 COPYCN.INPUT.TAPES.
 NEWIND.TAPES.
 WFLEC(430)
 LGO(PL=60000)
 WFLEC(1)
 *EOR
 *IU ZERNIKE
 *I GDL.261
 I ZERN = 0
 *I GDL.315
 I ZERN = 0
 *I SOU77CY1.165
 C = 23 APPLY UP TO 24 ZERNIKES IN UNITS OF WAVES. REA:IS ZFRNS
 *I GDL.24
 LOGICAL FRINGE
 *D GDL.295.SOU77CY1.167
 C /16 /17 /18 /19 /20 /21 /22 /23 /
 x.160.170.180.190.200.210.365.230).IFLOW
 *D GDL.325.SOU77CY1.169
 C /16 /17 /18 /19 /20 /21 /22 /23 /
 x.160.170.180.190.200.210.365.230).IFLOW
 *I GDL.327
 C.....
 C APPLY ZERNIKE
 C.....
 230 I ZERN = I ZERN + 1
 IF (.NOT.(INT)) GO TO 244
 FRINGE = .FALSE.
 DO 244 I=1,24
 248 P(I) = 0.

```

      DO 240 I=1,35
240 PFRNG(I) = 0.
      READ (5,ZFRNS)
      DO 239 I=1,35
239 IF (PFRNG(I).NE.0.) FRINGE=.T.
      IF (.NOT.FRINGE) GO TO 241
      WRITE (6,245)
245 FORMAT (/5X,'FRINGE COEFFICIENTS BEING CONVERTED TO 500 ORDER.*/)
      P(1) = 0.
      P(2) = PFRNG(1)
      P(3) = PFRNG(2)
      P(4) = PFRNG(3)
      P(5) = PFRNG(4)
      P(6) = PFRNG(5)
      P(7) = PFRNG(6)
      P(8) = PFRNG(7)
      P(9) = PFRNG(8)
      P(10) = PFRNG(10)
      P(11) = PFRNG(9)
      P(12) = PFRNG(11)

      P(13) = PFRNG(12)
      P(14) = PFRNG(16)
      P(15) = PFRNG(17)
      P(16) = PFRNG(13)
      P(17) = PFRNG(14)
      P(18) = PFRNG(18)
      P(19) = PFRNG(19)
      P(20) = PFRNG(25)
      P(21) = PFRNG(26)
      P(22) = PFRNG(15)
      P(23) = PFRNG(24)
      P(24) = PFRNG(35)
      IFRTST = 0
      DO 246 K=20,23
246 IF (PFRNG(K).NE.0.) IFRTST = 1
      DO 243 K=27,34
243 IF (PFRNG(K).NE.0.) IFRTST = 1
      IF (IFRTST.FI.1) WRITE (6,247)
247 FORMAT (/5X,'WARNING - FRINGE COEFFICIENTS OF ORDER 20 THROUGH 23*,
      C * AND 27 THROUGH 34 ARE IGNORED*/)
241 DO 242 I=1,24
242 PZSAVE(I,I/ZFRN) = P(I)
      PZSAVE(25,I/ZFRN) = 0
244 CALL ZFRN(PZSAVE(25,I/ZFRN),PZSAVE(I,I/ZFRN))
      IGNU = 1
      GO TO 999

*U GOL.27
      DIMENSION IPLTS(50),PZSAVE(25,10),P(24),PFRNG(35)
*I GOL.33
      DATA P,PFRNG/24*0.,.35*0./ , 00 / 5. /
*I GOL.243
C
      NAMELIST /ZFRNS/ 40,P,PFRNG
C
C      R0 = RADIUS OVER WHICH ZERNIKES ARE VALID.
C      P = ARRAY /ZERNIKE COEFFICIENTS.
C      PFRNG = ARRAY FRINGE ZERNIKE COEFFICIENTS (CONVERTED TO P IN GOL).
*I LWRP1.384
      SUBROUTINE ZFRN(40,P)
      LEVEL 2.CM

```

```

COMMON /MELT/ CU(32/68),CFIL(16512),X(128),WL,NPTS,NPY,DRX,DRY
COMPLEX CFIL
DIMENSION P(24)
IF(R0.F0.0.) GO TO 70
DO 100 IY=1,NPY
  J1 = (IY-1)*NPTS
  YS0 = X(IY)**2
  DO 100 IX=1,NPTS
    XS0 = X(IX)**2
    INDX = IX + J1
    R = SQRT(XS0+YS0)
52 THET = ATAN2(X(IY),X(IX))
    W = ANJ(1)(W/R0,1.)
    CT = COS(THET)
    C2T = COS(2.*THET)
    C3T = COS(3.*THET)
    C4T = COS(4.*THET)
    C5T = COS(5.*THET)

    ST = SIN(THET)
    S2T = SIN(2.*THET)
    S3T = SIN(3.*THET)
    S4T = SIN(4.*THET)
    S5T = SIN(5.*THET)
    R2 = R**2
    R3 = R*R2
    R4 = R*R3
    R5 = R*R4
    R6 = R*R5
    R8 = R2*R6
    R10 = R2*RA
    DEL = P(1) + P(2)*R*CT + P(3)*R*ST
A    + P(4)*(2.*R2-1.)
B    + P(5)*R2*C2T + P(6)*R2*S2T
C    + P(7)*(3.*R3-2.*R)*CT + P(8)*(3.*R3-2.*R)*ST
D    + P(9)*R3*C3T + P(10)*R3*S3T
E    + P(11)*(6.*R4-8.*R2+1.)

F    + P(12)*(4.*R4-3.*R2)*C2T + P(13)*(4.*R4-3.*R2)*S2T
G    + P(14)*R4*C4T + P(15)*R4*S4T
H    + P(16)*(10.*R5-12.*R3+3.*R)*CT
I    + P(17)*(10.*R5-12.*R3+3.*R)*ST
J    + P(18)*(5.*R5-4.*R3)*C3T + P(19)*(5.*R5-4.*R3)*S3T
K    + P(20)*R5*C5T + P(21)*R5*S5T
L    + P(22)*(20.*R6-30.*R4+12.*R2-1.)
M    + P(23)*(70.*R8-140.*R6+90.*R4-20.*R2+1.)
N    + P(24)*(252.*R10-630.*R8+560.*R6-210.*R4+30.*R2-1.)
60 INDX = INDX*2
    DEL = DEL*2.*3.141592654
    COSD = COS(DEL)
    SIND = SIN(DEL)
    CURS = CUR(INDX-1)
    CUR(INDX-1) = CURS*COSD - CUR(INDX)*SIND
100 CUR(INDX) = CURS*SIND + CUR(INDX)*COSD
    WRITE (6,200) R0,P
200 FORMAT ('0ZERNIKE PHASE CORRECTION APPLIED WITH NORMALIZATION*
A * RADIUS OF *.G15.4 /* COEFFICIENTS USED P(1)-P(24)*,
B * ARE CONSISTENT WITH THE PHASE DUE TO THE NTH TERM BEING*//
C 20X.24H PHI(N) = 2*PI*P(N)*Z(N)//

```

```

D * Z(N) = RF(N)*.1H*.F(THETA) ( RF(N) NORMALIZED TO 1. AT R=1.0//
E (IX.5620.5)
RETURN
70 NOB = NPPTS*NPY
DO RN I=1,NOB
  II=I+I
  IIM1=II-1
  CUR(IIM1) = SQRT(CUR(II)**2+CUR(IIM1)**2)
90 CUR(II) = 0.0
WRITE(6,300)
300 FORMAT(//IX.0CU PHASE HAS BEEN SET TO ZERO IN SUBROUTINE ZERN*//)
RETURN
END
*EUR
TEST ZERNIKE ADDITION
SSTART WWL=0.00106, NCALL=2, DCAL=15., NNPTS=124,
IB=8, DDHX=0.0, NDRY=0.0, AMPGES=20.0, NGAUSS=0.0,
RESTR=TRUE., PLOTS=1.0, IN=5,
SYMTRC=FALSE., PHIRAD=0.0, SEND
ST = SIN(THET)
S2T = SIN(2.*THET)
S3T = SIN(3.*THET)
S4T = SIN(4.*THET)
S5T = SIN(5.*THET)
R2 = R**2
R3 = R**3
R4 = R**4
R5 = R**5
R6 = R**6
RH = R**6
R10 = R**10
DEL = P(1) * P(2)*R*C1 + P(3)*R*ST
A * P(4)*R2-1.)
B * P(5)*R2*C2T + P(6)*R2*S2T
C * P(7)*(3.*R3-2.*R)*CT + P(8)*(3.*R3-2.*R)*ST
D * P(9)*R3*C3T + P(10)*R3*S3T
E * P(11)*(6.*R4-0.*R2+1.)
F * P(12)*(4.*R4-3.*R2)*C2T + P(13)*(4.*R4-3.*R2)*S2T
G * P(14)*R4*C4T + P(15)*R4*S4T
H * P(16)*(10.*R5-12.*R3+3.*R)*CT
I * P(17)*(10.*R5-12.*R3+3.*R)*ST
J * P(18)*(5.*R5-4.*R3)*C3T + P(19)*(5.*R5-4.*R3)*S3T
K * P(20)*R5*C5T + P(21)*R5*S5T
L * P(22)*(20.*R6-30.*R4+12.*R2-1.)
M * P(23)*(70.*R6-140.*R4+40.*R2-20.*R2-1.)
N * P(24)*(252.*R10-630.*R8+560.*R6-210.*R4+30.*R2-1.)
60 IND2 = IND1**2
DEL = DEL**2*.3.141592654
COSD = COS(DEL)
SIND = SIN(DEL)
CURS = CUR(IND2-1)
CUR(IND2-1) = CURS*COSD - CUR(IND2)*SIND
100 CUR(IND2) = CURS*SIND + CUR(IND2)*COSD
WRITE(6,200) P0,P
200 FORMAT (07F9.4F PHASE CORRECTION APPLIED WITH NORMALIZATION*
A * RADIUS OF .015.4 /* COEFFICIENTS USED P(1)-P(2)*,
B * ARE CONSISTENT WITH THE PHASE DUE TO THE NTH TERM BEING*//
C 20X.24H PHI(I) = 2*PI*P(N)*Z(N)//
D * Z(N) = RF(N)*.1H*.F(THETA) ( RF(N) NORMALIZED TO 1. AT R=1.0//
E (IX.5620.5)

```

```

RETURN
70 NUN = NPTS*NNY
   DO 40 I=1,NUN
     II=I+1
     IIM1=II-1
     CUR(IIM1) = SQRT(CUR(II)**2+CUR(IIM1)**2)
90 CUR(II) = 0.0
   WRITE(A,300)
300 FORMAT(//10X,'CU PHASE HAS BEEN SET TO ZERO IN SUBROUTINE ZERN//')
RETURN
END
*EUR
TEST ZERNIKE ADDITION
$START WWL=0.00106, NCALL=2, DCAL=15., NNPTS=124,
  I=4, DPHX=0.0, DURY=0.0, AMPGES=20.0, NGAUSS=0.0,
  RESTRT=.TRUE., PLOTS=1.0, IN=5,
  SYMTC=.FALSE., PHIRAD=0.0, $END
TEST ZERNIKE ADDITION
$CONTROL IFLOW=4, $END
  APERTURE THE PLANE WAVE TO 10. CM.
$APERTURE DOUBT=10., DIN=0., $END
$CONTROL IFLOW=8, $END
  PLOT THE INITIAL PLANE WAVE
$PLOT $END
  INITIAL PLANE WAVE
$CONTROL IFLOW=23, $END
  APPLY SPECIED ZERNIKES
$ZERNS R0=5, P(4)=.1, P(5)=.1, P(6)=.1, $END
$CONTROL IFLOW=8, $END
  PLOT THE ZERNIKED PLANE WAVE
$PLOT $END
  ZERNIKED PLANE WAVE
$CONTROL IFLOW=23, $END
  REMOVE SPECIED ZERNIKES
$ZERNS R0=5, PFRNG(3)=-.1, PFRNG(4)=-.1, PFRNG(5)=-.1, $END
$CONTROL IFLOW=8, $END
  PLOT THE DEZERNIKED PLANE WAVE
$PLOT $END
  DEZERNIKED PLANE WAVE
$CONTROL IFLOW=9, $END
  RETURN TO MAIN
$START WWL=-1., $END
*EUR

```

To obtain source printouts of the SOQ code, the user must run the CDC update program. The compile file may be used as a source listing or if the user so desires he may run the Fortran compiler on the code to obtain a compiled version or listing along with any desired Fortran compiler options supported under the CDC NOS/BE system. The file output will contain the desired listings. The following job setup is include as a guide:

Job Card

Account Card

Attach, OLDPL, SOQ77128, ID=

Update, F.

FTN.

REFERENCES

1. Sziklas, E.A., et al., System Optical Quality Study Phase I - Problem Definition, AFWL-TR-73-231, Pratt & Whitney Aircraft, June 1974.
2. Sziklas, E.A., and Siegman, A.E., "Mode Calculations in Unstable Resonators with Flowing Saturable Gain. 2: Fast Fourier Transform Method," Applied Optics, Vol. 14, pp. 1874-1899, August 1975.
3. Siegman, A.E. and Sziklas, E.A., "Mode Calculations in Unstable Resonators with Flowing Saturable Gain. 1: Hermite-Gaussian Expansion," Applied Optics, Vol. 13, p. 2775, 1974.
4. Krupke, W.F. and Sooy, W.R., "Properties of an Unstable Confocal Resonator CO₂ Laser System," IEEE Journal of Quantum Electronics, Vol. QE-5, No. 12, p. 579, December 1969.
5. Kogelnik, H. and Li, T., "Laser Beams and Resonators," Proc. IEEE, Vol. 54, pp. 1312-1329, October 1966.
6. Siegman, A.E., "Stabilizing Output with Unstable Resonators," Laser Focus, May 1961.
7. Goodman, J.W., Introduction to Fourier Optics, McGraw-Hill, New York, 1968.
8. Sziklas, E.A. and Siegman, A.E., "Diffraction Calculations Using Fast Fourier Methods," Proc. IEEE, Vol. 62, pp.410-412, March 1974.
9. Higgins, R.J., "Fast Fourier Transform; An Introduction with Some Mini-computer Experiments," AJP, 44, 1976.
10. Biblarz, O. and Fuhs, A.E., "Laser Cavity Density Changes with Kinetics of Energy Release," AIAA Journal, Vol. 12, p. 1083, August 1974.
11. Fuhs, A.E., "Quasidisk Area Rule for Heat Addition in Transonic and Supersonic Flight Regimes," AFAPL-TR-72-10, Air Force Aero Propulsion Laboratory, WPAFB, Ohio, 1972.
12. Tsien, H.E. and Milton Beilock, "Heat Source in a Uniform Flow," Journal of the Aeronautical Sciences, p. 756, December 1949.
13. Siegman, A.E. and Miller, H.Y., "Unstable Optical Resonator Loss Calculations Using the Prony Method," Applied Optics, Vol. 9, p. 2729, 1970.
14. Siegman, A.E., An Introduction to Lasers and Masers, McGraw-Hill, New York, 1971.

15. Humphreys, W.W. and Wick, R.V., "Change in Optical Path Length Near a Hot Mirror Surface," Laser Digest, AFWL-TR-75-140, 1975, p. 9.
16. Bergland, G.D., "A Guided Tour of the Fast Fourier Transform," IEEE Spectrum 6, 1969.
17. Cooley, J.W., Lewis, P.A.W., Welch, P.D., "The Finite Fourier Transform," IEEE Trans. Audio Electroacoust., AU-17, 1969.

DATE
ILME