

AD-A103 286

UNITED TECHNOLOGIES CORP WEST PALM BEACH FLA
SYSTEM OPTICAL QUALITY USERS GUIDE, PART 3.(U)
MAR 80 J L FORGHAM, S S TOWNSEND

F/G 20/5

F29601-77-C-0025

UNCLASSIFIED

AFWL-TR-79-141-PT-3

NL

File
40.6
10-81



END
DATE
FILMED
10-81
DTIC

12 13
AFWL-TR-79-141, Pt. 3

2

LEVEL III

AD-E200 724

AFWL-TR-
79-141
Pt. 3

AD A103286

SYSTEM OPTICAL QUALITY USERS GUIDE,

Part 3 of 3

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

United Technologies Corporation
West Palm Beach, FL 33402

11 Mar 1980 12 71

Final Report, Jan - Jan 1980

DTIC
ELECTE

AUG 24 1981

B

10 F2101-77-C-0025
Approved for public release; distribution unlimited.

16 14

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

DTIC FILE COPY

81 5 20 056

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 has 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

FOR THE DIRECTOR

James L. Stapp

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

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 PLID = 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:

$$\phi(1,J) = \sum_k P_k \pi Z_k(1,J)$$

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

$$\text{COSROT} = \cos(\theta)$$
$$\text{SINROT} = \sin(\theta)$$
$$\text{XROT} = \text{XIN} \times \text{COSROT} + \text{YIN} \times \text{SINROT}$$
$$\text{YROT} = -\text{XIN} \times \text{SINROT} + \text{YIN} \times \text{COSROT}$$

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.115

DEL = CFL*3.14159264

*DELETE ZRNKE.125

C 2(X,22F FFI(N) = F1*F(N)*Z(N)//

*IDENT FIXCTR

*/ JITREC

*DELETE JITTER.25, JITTER.30

CF = 1./(FLCAT(NPTS)*C)

*IDENT RCTZRN

*/ GCL

*DELETE ZRNINFC.3

NAMelist /ZERNS/ FC,F,FFPAC,SIGMAY,NTERMZ,ZTHETA,FLIFX,FLIFY

*INSERT ZRNKE.5

C ZTHETA = THE CLOCKWISE ANGLE OF ROTATION OF THE DECOMPOSITION
C AXES ONTO THE SOG COORDINATE SYSTEM
C BEFORE CALCULATION OF THE ZERNKE POLYNOMIALS.
C IT IS INPUT IN DEGREES.

C FLIFX = 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.

FLIFX = 0.

FLIFY = 0.

*INSERT ZRNINFC.9

FZ2SV(IZERN,3) = ZTHETA*3.141593/180.

PZ2SV(IZERN,4) = FLIFY

FZ2SV(IZERN,5) = FLIFX

*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

SUBROUTINE ZERN(SIGMAY,NTERMZ,THETA,FLIFX,FLIFY,FC,F)

*INSERT ZRNKE.72

CCSROT = COS(THETA)

SINROT = SIN(THETA)

*DELETE ZRNKE.75

*DELETE ZRNKE.77

XIN = X(IX)

YIN = X(IY)

IF(FLIFY.CT..5) YIN=-YIN

IF(FLIFX.CT..5) XIN=-XIN

XRCT = XIN*CCSROT + YIN*SINROT

YRCT = -XIN*SINROT + YIN*CCSROT

IF(FLIFX.LT.-.5) YRCT=-YRCT

IF(FLIFY.LT.-.5) XRCT=-XRCT

XSC = XRCT**2

YSC = YRCT**2

*DELETE ZRNKE.40

THET = ATAN2(YRCT,XRCT)

*IDENT MORSUM

*INSERT SUMMARY.F15

C

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

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.EQ.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.EQ.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.EQ.1) WRITE(6,3035)

WRITE(6,4040) C2

GO TO 6000

6015 REWIND 57

WRITE(6,3035)

C

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

C

REWIND ISUMRY

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

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

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

WRITE(6,3040) C2,C3

GO TO 5000

5015 REWIND ISUMRY

WRITE(6,3035)

C

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

C

WRITE(6,3035)

REWIND 50

8000 READ(50,4005) IC1,C2

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

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

WRITE (6,4040) C2
CC TO SELL
FCIS NEWID 50
WRITE (6,3035)
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.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SYSTEM OPTICAL QUALITY USERS GUIDE Part 3 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 64
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 1473
1 JAN 73

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	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A	23
	CP

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECTION V APPENDICES
APPENDIX A
SOQ USERS GUIDE UPDATES
JANUARY 1977 TO JUNE 1978

INTRODUCTION

This appendix documents those changes made to the initial SOQ code between January 1977 and June 1978. The changes incorporated in the code are those that have become generally useful for the physical optics simulation problems which have been solved using the SOQ code. The Users Guide Updates are also prepared to clarify and correct the initial description of the SOQ code, as documented and delivered to AFWL on 1 March 1978, in the Preliminary SOQ Users Guide. This document supersedes previous written material on SOQ code documentation. The organization of the SOQ Users Guide Updates is

SECTION AI	<u>New Subroutines</u>
	1. Theory
	2. FORTRAN Updates
SECTION AII	<u>Code Changes/Corrections</u>
	1. Theory/Reason for Correction
	2. FORTRAN Updates
SECTION AIII	<u>Users Guide Corrections</u>
SECTION AIV	<u>SOQ Code Access</u>

AI. NEW SUBROUTINES

1. THEORY

a. Beam jitter -- Relative motion between optical elements, such as mirrors, induces time varying positional displacement of the optical field. The typical term for this phenomenon is beam jitter, and the principle effect is to broaden the time-averaged effective beam illumination area, while reducing the time averaged intensity.

Beam jitter is both a near field and far field concern. Jitter in optical trains can overload apertures or cause energy deposition on areas outside the normal propagation path as well as cause a deterioration in the peak on-axis irradiance and integrated spot power.

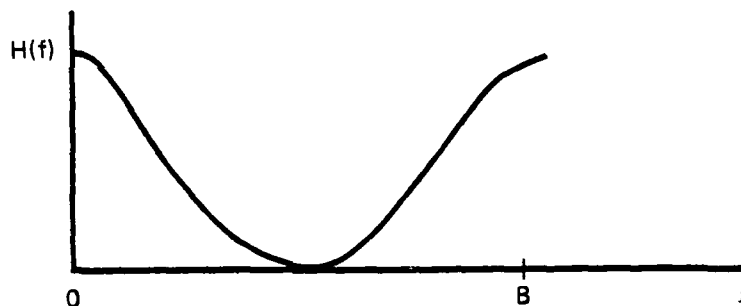
The time-averaged effect of beam jitter may be modeled as the convolution of the intensity profile with an appropriate probability density function (PDF) for the jitter statistics. The current SOQ model assumes that the jitter PDF is Gaussian with known mean and variance. The model allows the user to specify the Gaussian parameters and, for the selected beam jitter analysis location, to determine the near field and/or far field effect of beam jitter.

The following is a brief description of the analytical and SOQ Fortran implementation of beam jitter calculations:

b. Relevant formalism -- The effect on the beam may be found by convolution of the Gaussian jitter probability density function with the SOQ predicted intensity distribution:

$$I'(x,y) = \iint_{-\infty}^{\infty} I(x',y') J(x-x', y-y') dx' dy' \quad (A1)$$

The 1-D Fourier transform of the Gaussian function looks like:



2. FORTRAN UPDATES

The jitter model can be called in two ways. Each assumes that the jitter variance is the product of a jitter angle and the propagation distance from the jitter source.

$$\sigma = \theta_J \cdot Z \quad (A2)$$

θ_J = Jitter angle (1σ , in microradians)

Z = Distance from jitter source (in cm)

When the far field model is called from QUAL, the jitter angle has been incorporated into namelist QL0T while the propagation distance is the focal length found in QUAL. The jittered intensity is returned to array CU as a phaseless field so it can be plotted, or written to a permanent file.

The other method of activating the jitter model is to call the near field jitter model from GDL with IFLOW = 23. For this model both angle and jitter distance are entered in namelist JITTER.

Namelists modified:

Far Field

QLOT: SIGANG (rad) is added to specify the jitter angle

Near Field

JITTER: Contains -
JITANG (urad)
JITDIS - Jitter distance

AII. CODE CHANGES/CORRECTIONS

1. THEORY/REASON FOR CORRECTIONS

a. Bare resonator calculations -- The SOQ resonator/optical-train calculation code may be used to simulate, in Cartesian coordinates, bare resonators. This added option is frequently used in the initial simulation studies of a resonator or a class of resonators.

The bare resonator optical configuration may be compared to its geometric counterpart using the SOQ code by simply invoking the IBARE option and associated updates now contained in the fundamental code. The fundamental approach in bare resonator calculations on the SOQ code is to allow the user to use the same input and code for bare, semibare and loaded cavity calculations. Various options under the bare cavity calculations have been incorporated and are now described as input values for IBARE in Namelist START.

IBARE = 0 (Default)

Loaded cavity calculations are performed as usual following the standard input which the user has supplied.

IBARE = 1

Using the same input, the user will now perform bare cavity calculations in which the resonator is normalized to 1W of circulating power. Mirrors are defaulted to have 100 percent reflectivity, and no power dependent or flux dependent distortion. The SOQ output is modified to printout the resonator eigenvalue.

IBARE = 2

Semibare resonator calculations are performed in which the user can perform bare resonator calculations that include optical aberrations generated by a flowing saturable gain medium. These aberrations may strongly effect mode shape/phase. This option provides a convenient method of studying their perturbational effect on the bare cavity mode.

For the semibare option, an additional update has been included in which the namelist MIRROR user may specify the desired power at each of the resonator mirrors. This allows the user of the semibare updates to apply mirror distortions as though the bare cavity mode had a significant power level. Specification of the DESIPW value in namelist MIRROR to some value other than 0.0 will cause the field incident on the mirror to be scaled to that power specified by the numerical value of DESIPW. Appropriate mirror distortions will be applied at the desired power level. The field leaving the mirror will then be rescaled to its incident power level. Subsequent calculations are done as specified by the user typical namelist input.

An additional variation is allowed in which the parameter FLAG of cavity input namelist CAVTY2 can be used to execute a resonator with loaded gain, but no fixed phase perturbation in cavity. The input would correspond to FLAG = 0; IBARE = 0. Usual loaded resonator calculations are performed with mirror distortions as specified by the user.

All of the above variations of cavity/resonator calculations may be run from the standard loaded cavity input.

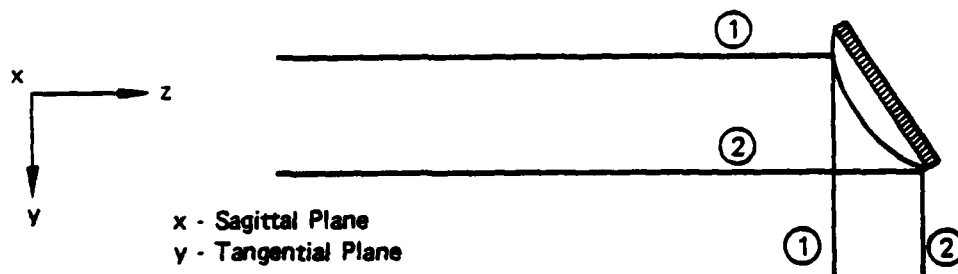
At the rear of this section are Fortran listings of the code updates which have been included in the basic Cycle III SOQ code previously documented.

b. Mirror non-normal incidence angle -- In many optical train calculations the propagating optical field is incident on the mirrors in a nonnormal manner. Since, in general, the mirror surface may have a spherical figure, the field leaving the mirror will exhibit phase front aberrations introduced by non-normal incidence of the field on the curved surface.

The SOQ MIRROR subroutine has been modified to incorporate an astigmatic aberration due to the nonnormal incidence on a spherical surface. The following is a brief description of the generation of the astigmatic aberration applied.

Astigmatism in Resonator:

General astigmatism is introduced when a wavefront is incident on a spherical (parabolic) surface in a nonnormal manner. This aberration occurs at each spherically-distorted turning flat, for example.



$$\frac{1}{S} + \frac{1}{S'_s} = -2 \frac{\cos \phi}{R_c}$$

$$\lim_{S \rightarrow \infty} S'_s = -\frac{R_c}{2 \cos \phi}$$

$$\Delta \theta_{SOQ_s} = \frac{2\pi}{\lambda} \left(\frac{x^2}{2S'_s} \right)$$

ϕ = Incident angle

S = Object distance

R_c = Mirror surface curvature (spherical)

S'_s = Sagittal plane effective curvature

Thus $2S'_s$ is the resultant phase curvature being imposed on the beam. A cylindrical mirror can be used to model this with

$$R_{c_{S'_s}} = 2S'_s = -\sqrt{2} R_c \text{ (neg since } R_c \text{ is convex) for } \phi = 45^\circ \quad (A3)$$

Therefore, to represent the astigmatism introduced in the x-plane by a spherically-distorted turning flat, a cylindrical mirror is employed with a radius of curvature

$$R_{c_{S'_s}} = -\sqrt{2} R_c$$

R_c is the power induced radius of curvature which is input or determined by the SOQ code.

Similarly, the tangential plane is described by

$$\Delta \theta_{SOQ_T} = \frac{2\pi}{\lambda} \left(\frac{y^2}{2S'_T} \right)$$

$$\begin{aligned}
 S_T &= \frac{-R_C \cos \phi}{2} \\
 &= \frac{-R_C}{2\sqrt{2}} \text{ for } \phi = 45^\circ
 \end{aligned}
 \tag{A4}$$

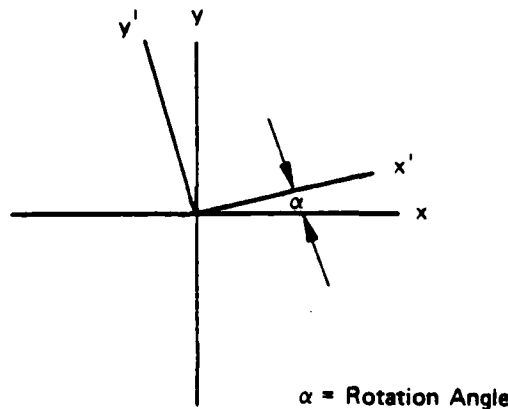
The new mirror subroutine including astigmatic effects has the form

$$\begin{aligned}
 \Delta \theta_{\text{SOQ}} &= \frac{2\pi}{\lambda} \left[\frac{x^2}{2S'_S} + \frac{y^2}{2S'_T} \right] \\
 S'_S &= \frac{R_C}{2 \cos \phi} & S'_T &= \frac{R_C \cos \phi}{2}
 \end{aligned}
 \tag{A5}$$

The only additional input change is to the MIRROR routine namelist which is expanded to include the variable PHIAST, the beam incidence angle in degrees (default is PHIAST = 0).

c. Beam rotation -- The mirror model has been updated to describe beam rotation introduced by optical elements which are oriented in a skewed fashion. Many examples of this type of orientation are encountered in resonators and optical trains. The principle effect of skewed, or out-of-plane, orientation is to convolve or smooth the mirror distortion-induced aberrations over the total number of optical elements.

Rotation of the beam is accomplished by analytically rotating the mirror with respect to the beam, rather than rotating the beam within the mesh and then applying the mirror. By rotating the mirror with respect to the beam two modeling advantages result: First, analytical rotation of the mirror with respect to the beam is accomplished with no interpolation loss of information. Second, since the rotation is analytical, computer time is saved by not having to evaluate the field numerically. The following describes the rotation equations used in the code. The following sketch shows a base and rotated system.



Since,

$$x = x \cos \alpha + y \sin \alpha \quad (A6)$$

$$y = -x \sin \alpha + y \cos \alpha \quad (A7)$$

Then,

$$\Delta \phi = \frac{2\pi}{\lambda R_T} \left\{ \frac{(x')^2}{\cos \alpha} \right\} + (y')^2 \cos \alpha \quad (A8)$$

Here,

(x,y) are the SOQ coordinates

(x',y') are the transformed (rotated) coordinates

The SOQ field is modified as

$$CU_{OUT} = A \exp (\Delta \phi) CU_{IN} \quad (A9)$$

where A represents the completed transmittance effects included in mirror.

The variable added to the SOQ MIRROR namelist input is PHIROT, which is the beam rotation angle in degrees. The default value is PHIROT = 0.0.

2. FORTRAN UPDATES

The attached printouts contain a listing of the updates, denoted as ROT, used to effect these changes.

IDENT ROT

*IDENT ROT

*DELETE C10ASTG.1

ATOP(3.4),XSCR(4),AHC(14.20.9),TITLE3(20),XUPADD(4),

*DELETE C10FLA.1

DIMENSION ID(4(5.24),IGUL(99),AHC(14.20.9),CFEL(16384),IJSK(4.9),

*DELETE C10ASTG.2,C10ASTG.3

DATA WANULS,DOUZY,DINY,PHIAST,PHIROT /500./

*DELETE C10ASTG.4

A DELTA,DISTF,DOUZY,DINY,WANULS,PHIAST,PHIROT,DESIPW

*INSERT C10ASTG.5

C PHIROT IS THE BEAM ROTATION ANGLE AT THAT STATION-- DEG

C DESIPW IS THE DESIRED POWER LEVEL AT THAT STATION

*INSERT C10ASTG.6

AHC(13,IMIR.2)=PHIROT

AHC(14,IMIR.2) = DESIPW

22 DESIPW = AHC(14,IMIR.2)

*DELETE C10ASTG.8

X DISTF,WANULS,RYOUT,RYIN,PHIAST,PHIROT,DESIPW)

```

*INSERT MIRHOR.2H
  PHIROT=PHINT
*INSERT CIOASTG.12
  PHIROR=-PHIRUT*PI/180.
  PHIROT=0.0
  WRITE (A.86) PHIROR
  SINPR=SIN(PHIROR)
  COSPR=COS(PHIROR)
*DELETE CIOASTG.15
  XPRM=X(J)*COSPR*X(I)*SINPR
  YPRM=-X(J)*SINPR*X(I)*COSPR
  PHASE=AKY*((XPRM**2/RMSAG)+(YPRM**2/RMTAN))-AKY*DELL
*DELETE CIOASTG.24
  XPRM=X(J)*COSPR*X(I)*SINPR
  YPRM=-X(J)*SINPR*X(I)*COSPR
  PHASE=AKY*((XPRM**2/RMSAG)+(YPRM**2/RMTAN))
*DELETE MIRHOR.84
  PHIR=(PHIAST*PI)/180.
  RMSAG=RUC/COS(PHIR)
  RMTAN=WOC/COS(PHIR)
  PHIROR=-PHIROT*PI/180.
  PHIROT=0.0
  SINPR=SIN(PHIROR)
  COSPR=COS(PHIROR)
*DELETE MIRHOR.91
  XPRM=X(J)*COSPR*X(I)*SINPR
  YPRM=-X(J)*SINPR*X(I)*COSPR
  PHIMIR=AKY*((XPRM**2/RMSAG)+(YPRM**2/RMTAN))
*INSERT MIRHOR.10H
  WRITE (A.86) PHIROR
  WRITE (A.420) RMSAG,RMTAN
  DO      FORMAT(20X,' MIRHOR ROTATION = ',G12.5,'RADS')

```

AIII. USER'S GUIDE CORRECTIONS

1. 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. The FUHS modeling includes the assumption generally made for supersonic flow and assumes the heat release has only a small perturbative effect on the flow.

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

The usual continuity, momentum, and energy equations for steady flow with heat addition are used as the basis for the analysis:

$$\begin{aligned}
 \text{Continuity:} \quad & \nabla \cdot (\rho \vec{u}) = 0 \\
 \text{Momentum:} \quad & \rho \frac{D\vec{u}}{Dt} + \vec{\nabla} p = 0 \\
 \text{Energy:} \quad & \nabla \cdot \rho \vec{u} h + \frac{\vec{u}^2}{2} = q
 \end{aligned}$$

These are linearized assuming

$$\begin{aligned}\rho &= \rho_{\infty} + \rho' \\ p &= p_{\infty} + p' \\ \vec{u} &= \hat{i} (U + u') + \hat{j} v'\end{aligned}\tag{A10}$$

Resulting in

$$\text{Continuity: } \rho_{\infty} u'_x + \rho_{\infty} v'_y + U \rho'_x = 0\tag{A11}$$

$$(u'_x = \frac{\delta}{\delta x} u'; \text{ etc})\tag{A12}$$

$$\begin{aligned}\text{Momentum: } \rho_{\infty} \vec{U} u'_x + p_x &= 0 \\ \rho_{\infty} U v'_x + p_y &= 0\end{aligned}\tag{A13}$$

$$\text{Energy: } \frac{p_{\infty} U}{\gamma - 1} \frac{\delta}{\delta x} \left(\frac{p'}{p_{\infty}} - \frac{\gamma \rho'}{\rho_{\infty}} \right) = q\tag{A14}$$

The solution to these equations was first shown by Tsien and Bieloeh, (Ref. 12), resulting in the following equations for a heat source q in supersonic heat addition.

$$u' = \frac{-(\gamma - 1)q}{2\gamma p \beta} \delta(x - \beta y)\tag{A15}$$

$$v' = \frac{(\gamma - 1)q}{2\gamma p} \delta(x - \beta y)\tag{A16}$$

$$p' = \frac{(\gamma - 1)Mq}{2a^3 \beta} \delta(x - \beta y) - \frac{(\gamma - 1)q}{a^2 U} \delta(y) I(x)\tag{A17}$$

Where,

$$x = \beta y$$

Defines a Mach line

$$\beta = \sqrt{M^2 - 1}$$

$$a = \frac{U}{M}$$

Speed of Sound

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

For a small volume, the heat addition is $q = h(x,y) dx dy$. The effects of all sources are then added; for example,

$$U = - \frac{(\gamma-1)}{2\gamma p\beta} \iint h(x,y) \delta(x-\beta y) dx dy$$

$$= - \frac{(\gamma-1)}{2\gamma p\beta} \int_0^s h(x=\beta y) \sin \mu ds \quad (A18)$$

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

$$x = s \cos \mu \quad y = s \sin \mu \quad (A19)$$

By the Faltung theorem, for Fourier transforms, this can be written

$$I'(x,y) = F^{-1} \left\{ F[I(x,y)] \cdot F[J(x,y)] \right\} \quad (A20)$$

The Fourier transform of the intensity is performed by the FFT, while the Fourier Transform for Gaussian density functions can be found analytically as

$$F \left\{ \frac{1}{2\pi\sigma^2} \exp \left[-\frac{(x^2+y^2)}{2\sigma^2} \right] \right\} = \exp \left[-2\pi\sigma^2 (f_x^2 + f_y^2) \right] \quad (A21)$$

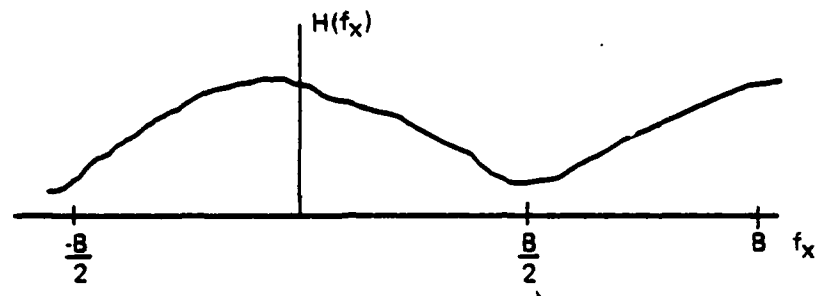
From sampling theory, the discrete values of f_x and f_y can be found since

$$\Delta f = \frac{1}{L} \quad (A22)$$

where

L is the width of the SOQ calculation region (DCALC)

$f_x(I)$ is then $(I-1)\Delta f$. Recall from the discussion in FOURT, the DC value is returned in the first position and the last half of the transformed data are really negative frequency information shifted by one period, illustrated below in one dimension.



where

$$B = \frac{1}{\Delta x}$$

Δx = Sampling rate in real space

The equation for density change is, therefore,

$$\frac{\Delta \rho}{\rho} = \frac{1}{\rho} \left[\frac{(\gamma - 1)M}{2a^3 \beta} \int_0^S h(x, y) \Big|_{x=\beta y} \sin u \, ds \right. \\ \left. \frac{(\gamma - 1)}{a^2 U} \iint dx' dy' h(x', y') \delta(y - y') I(x - x') \right] \quad (A23)$$

The first term describes the compression waves along the streamlines due to heat addition, while the second describes the wake resulting from those compression waves.

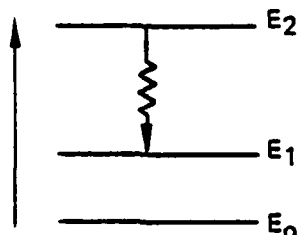
The heat release $h(x, y)$ for a laser can be written:

$$h(x, y) = C \int_{x'_{NEP}}^x \Delta I(x', y) e^{-\frac{(x - x')}{R\tau}} \quad (A24)$$

where τ is the time constant for the depopulation of the lower laser level. If the depopulation were instantaneous ($\tau \rightarrow 0$), the heat release would be proportional to the intensity, since every molecule emitting a photon would then immediately relax to the ground state with an accompanying increase in translational energy. It has been shown that the above equation for the heat release can be used in all regions of the cavity with only small error introduced.

The constant c can be found by conservation of energy as shown following.

Consider the following 3-level molecule:



The quantum efficiency η is defined as the ratio of the energy out divided by the total energy available, so for the gain/phase segment under consideration.

$$\eta = \frac{(\text{no. of molecules})(E_2 - E_1)}{(\text{no. of molecules})(E_2 - E_0)} = \frac{\Delta P}{\Delta H + \Delta P} \quad (\text{A25})$$

Where

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

the above expression can be inverted to give

$$\Delta H = \left(\frac{1 - \eta}{\eta} \right) \Delta P \quad (\text{A26})$$

with

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

and

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

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

$$\begin{aligned} \Delta H &= c \Delta z \int_0^\infty dy \int_0^\infty dx \int_0^x \Delta I(x', y) e^{-(x - x')/U\tau} \, dx' \\ &= c \Delta z \int_0^\infty dy \int_0^\infty dx \int_0^\infty I(x - x') \Delta I(x', y) e^{-(x - x')/U\tau} \, dx' \end{aligned} \quad (\text{A27})$$

Where, recall

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

So,

$$\begin{aligned} \Delta H &= c \Delta z \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dx' \Delta I(x', y) \int_{-\infty}^{\infty} dx I(x - x') e^{-(x - x')/U\tau} \\ &= c \Delta z \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dx' \Delta I(x', y) \int_{x'}^{\infty} dx'' e^{-x''/U\tau} \\ &= c \Delta z \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dx' \Delta I(x', y) \left(\frac{1}{1/U\tau} \right) \end{aligned} \quad (A28)$$

Or,

$$\Delta H = c(\Delta z) U\tau \Delta P$$

Or,

$$c = \left(\frac{1 - \eta}{\eta} \right) \frac{1}{U\tau \Delta z}$$

Since the numerical kinetics routine returns information about the wake region itself and not just the heat addition terms, this information must be the data used. Thus, for the analytical kinetics model, one must find the value for the wake integral:

$$\begin{aligned} w(x, y) &= \int_{-\infty}^x dx' h(x', y) = c \int_{-\infty}^x dx' \int_{-\infty}^x dx'' \Delta I(x'', y) e^{-(x' - x'')/U\tau} \\ &= c \int_{-\infty}^{\infty} dx' I(x - x') \int_{-\infty}^{\infty} dx'' I(x' - x'') \Delta I(x'', y) e^{-(x' - x'')/U\tau} \\ &= c \int_{-\infty}^{\infty} dx'' \Delta I(x'', y) \int_{-\infty}^{\infty} dx' I(x - x') I(x' - x'') e^{-(x' - x'')/U\tau} \\ &= c \int_{-\infty}^{\infty} dx'' \Delta I(x'', y) I(x - x'') \int_{x''}^x dx' e^{-(x - x'')/U\tau} \end{aligned} \quad (A29)$$

So,

$$w(x,y) = c \int_0^x dx'' \Delta I(x'',y) U \tau (1 - e^{-(x-x'')/U\tau}) \quad (A30)$$

So, recalling

$$c = \left(\frac{1-\eta}{\eta} \right) \frac{1}{U\tau\Delta z} \quad (A31)$$

And

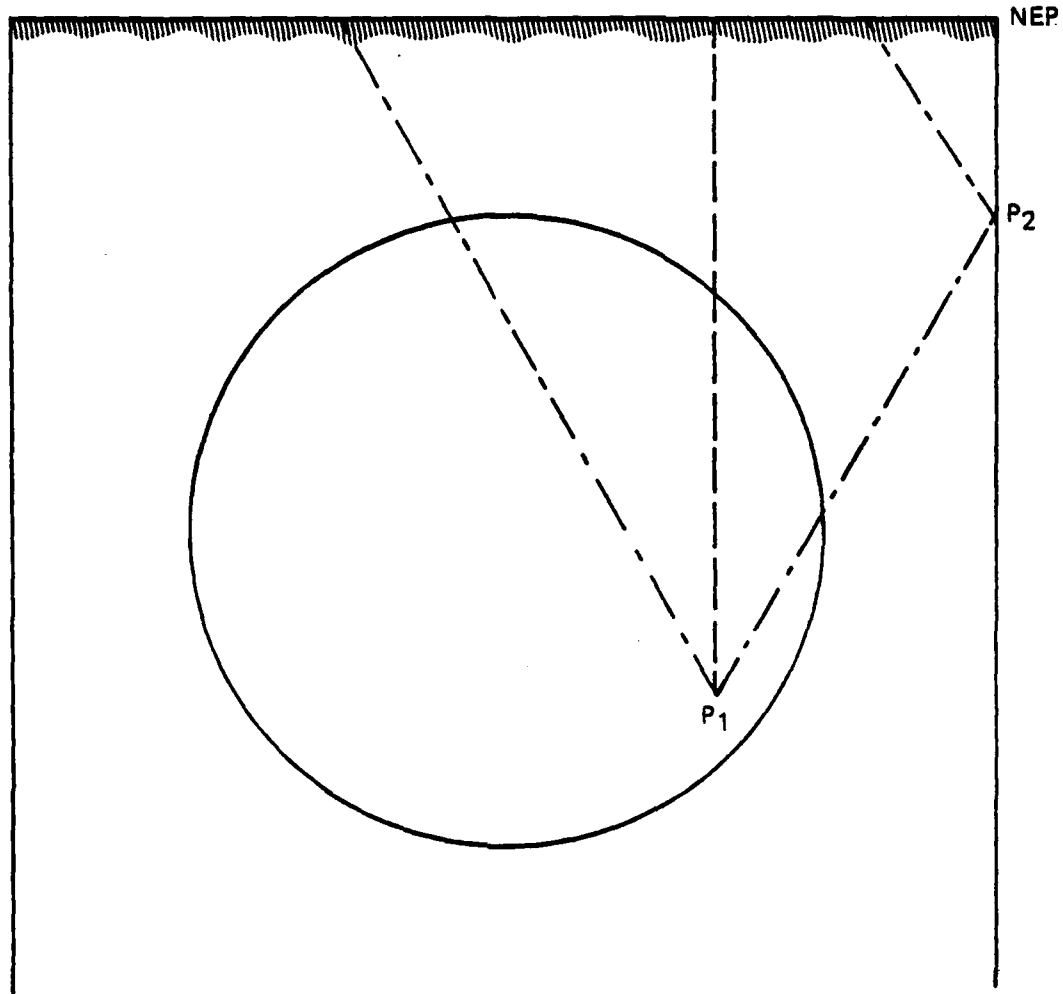
$$\Delta I(x'',y) = 2 \left(\frac{1-G}{1+G} \right) \text{PPD} \quad \text{from SIMPGG}$$

$$w(x,y) = \frac{2}{\Delta z} \left(\frac{1-G}{1+G} \right) \left(\frac{1-\eta}{\eta} \right) \int_0^x dx' \text{PPD}(x',y) (1 - e^{-(x-x')/U\tau}) \quad (A32)$$

Now both numerical and analytical kinetics models return the same array, namely the value of the wake integral throughout the cavity. The effect of heat release due to lower level depopulation can be calculated without regard to the particular kinetics model chosen. The Fuhs effect is calculated in the following manner:

$$H(I,J) = \frac{1}{\Delta x} \int_{x(I-1)}^{x(I)} h(x,y) dx = \frac{w(x(I)) - w(x(I-1))}{\Delta x} \quad (A33)$$

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 the following diagram:



The contribution at P_1 due to reflection 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 .

The phase shift is found using the Gladstone-Dale law

$$n = 1 + C_0 \quad (A34)$$

The phase change $\Delta\phi$ is

$$\Delta\phi = \frac{2\pi}{\lambda} (\Delta n) (\Delta z) = \frac{2\pi}{\lambda} \left(\frac{C\Delta\rho}{\rho_0} \right) \rho_0 \Delta z \quad (A35)$$

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

AIV. SOQ CODE ACCESS

1. SOURCE CODE

The following listing represents the source code necessary to update the SOQ to include the corrections and modifications described on the preceding pages.

1000 1000

```

*INVENT IBAKE
*INSERT GDL.531
  IF (IBARF.EQ.1) GO TO 850
  WMIR = 1.0
  DELTA = 0.0
  DISTF = 0.0
  WRITE(6,660) IBARF
660  FORMAT(/2X,10H*****. * IBAKE =,I2,. * DELTA AND *.
  C * DISTF SET TO ZERO AND WMIR SET TO 1. *.10H*****/)
850  CONTINUE
*INSERT APP24.20
  WRITE(7) (CU(IZ),IZ=1,NOM)
  REWIND 7
*DELETE      GDL.704,GDL.705
*DELETE      GDL.690,GDL.702
  IF (ICAV.EQ.0) IBAKE=1
  IF (IBARF.EQ.0) GO TO 691
  WRITE(29) (CFFL(IZ),IZ=1,NOM).X
  REWIND 29
  DO 684 I=1,NOM
684   CU(I) = CFFL(I)
  DO 684 I=1,NPTS
6841  X(I) = XK(I)
  DIHMM = ARC(1.1,1)/2.
  CALL APWTH(DIHMM,0.0,0.0,0.0,0.0)
  POW=0.
  READ(29) (CFFL(IZ),IZ=1,NOM).X
  REWIND 29
  DO 684 I=1,NOM
684   POW = POW + CU(I) * CONJG(CU(I))
  POW = POW * (XK(2) - XK(1))**2*(NPTS/NBY)
  WRITE(6,687) POW
687   FORMAT(5X,. * --- POWER IN FEEDBACK NORMED TO UNITY BY *.
  XE15.7,/)
  SQTPOW = SQRT(POW)
  DO 686 I=1,NOM
686   CFFL(I) = CFFL(I) / SQTPOW
*DELETE      GDL.822,GDL.823
  IF (IBARF.EQ.0) GO TO 1002
  IF (INIT.AND..NOT.RESTART) GO TO 87
  C ***** CALCULATE EIGENVALUE *****
  IF (PPWK.GT.0.001) PPWK = .001
  EIG = SQRT(1.-1000.*PPWK)
  WRITE(6,86) EIG
86   FORMAT(20X,. * ESTIMATED EIGENVALUE = *.E15.7,/)
  GO TO 1003
87   WRITE(6,88)
88   FORMAT(20X,. * FIRST PASS INPUT POWER NOT UNITY, EIG NOT ESTD.,. * /)
  GO TO 1003

```

```

1002 CONTINUE
      CALL REGAIN(NCT,NITER)
1003 IF(ICKR.EQ.0) GO TO 565
*DELETE      GDL.428.GDL.438
*DELETE CAVITY.3
      X ZLI,ZLO,IRARE)

*INSERT CAVITY.141
      IF(IRARE.NE.0) GO TO 109
*DELETE CAVITY.143
      109 WRITE (7) (CH(IZ),IZ=1,MUT)
*DELETE CORR1.47
      CALL DENSITY(FLAG,RHO,XLEN,YLEN,UCZ,NAA,NYA,1,IN,NNSYM,IBARE)
*INSERT SQRTCYL.10
      IF(IRARE.GT.0) GO TO 12
*INSERT CAVITY.217
      GO TO 11
      12 CG(I7)=CMPLX(COS(PHIM),SIN(PHIM))
*DELETE DENSITY.2
      SUBROUTINE DENSITY(FLAG,RHO,XLEN,YLEN,ZSLAR,NPX,NPY,IF,IN,NNSYM,
      X IRARE)
*INSERT DENSITY.105
      IF(IRARE.EQ.1) RETURN
      IF(LAG.GT.0) GO TO 12
      *WRITE(6,13)
      13 FORMAT(//10X,5H*** **FLAG = 0. IN DENSITY,5H ***/
      A 15X, **ALL JPOS SET TO 0.0*///)
      RETURN
      12 CONTINUE
*INSERT LROPI.2
      DATA IRARE/0/
*INSERT GDL.47
      DATA DESIPW/0./
*INSERT LROPI.1
      COMMON /HARES/ IRARE
*INSERT GDL.22
      COMMON /HARFS/ IRARE
*DELETE LROPI.3
      X ,IRARE,PLOTS

C
C      IRARE IS FLAG FOR LOADED, HARE, OR SEMI-HARE CAVITY
C      = 0 FOR LOADED RESONATOR (DEFAULT VALUE)
C      = 1 FOR HARE RESONATOR (UNITY GAIN,0 PHASE CHANGE)
C      = 2 FOR SEMI-HARE RESONATOR (UNITY GAIN,DENSITY PHASE CHANGE)
*INSERT GDL.430
      IDIR(5,ICAV) = IRARE
*INSERT GDL.437
      X,IDIR(5,ICAV)
*DELETE GDL.439
      X PROPAGATING PARAMETER ,I2/* IRARE= *,I3)
*DELETE GDL.444
      X RESTHT,IDIR(4,ICAV),ZLI(ICAV),ZLO(ICAV),IDIR(5,ICAV))
*DELETE GDL.530
      IF(.NOT.INIT) GO TO 22
      DESIPW = 0.0
*INSERT GDL.245
      NOH = NPTS*NPY
*DELETE MIRROR.24
      70 IF(DESIPW.EQ.0.0) GO TO 360
      NOH = NPY*NPTS
      NOH2 = NOH*2
C *** FIND INCIDENT POWER

```

IDENT NAME

```

PPWIN = 0.0
DO 355 IZ=1,NOR2,2
355 PPWIN = PPWIN + CUW(IZ)**2 + CUW(IZ+1)**2
PPWK = PPWIN*(X(2)-X(1))**2*(NPTS/NPY)/1000.
IF(NRFG.EQ.1.OR.NREG.EQ.2) PPWK=PPWK/WNO4**2
TRANS = SQRT(DFS(PW/PPWK))
C ***** SCALE THE BEAM TO THE DESIRED POWER.
DO 356 IZ = 1,NOR2
356 CUW(IZ) = CUW(IZ)*TRANS
WRITE(6,4010) DESIPW,PPWK,TRANS
4010 FORMAT(/5X,42H THE FIELD HAS BEEN SCALED TO DESIRED POWER/
X 8X,12H DESIPW =,G12.4/8X,12H PPWK =,G12.4/
X 8X,12H TRANS =,G12.4/)
DESIPW = TRANS
360 IF(ABS(ANX).LE.0.000100.AND.ABS(ANY).LE.0.000100) GO TO 71
*DELETE C10ASTG.7
3 AHC(11,IMIR,4),ARC(12,IMIR,2),ABC(13,IMIR,2),DESIPW)
IF(DESIPW.FU.0.0) GO TO 24
C ***** SCALE THE FIELD BACK DOWN.
DO 358 IZ=1,NOR
358 CU(IZ) = CU(IZ)/DESIPW
WRITE(6,4000)
4000 FORMAT(/5X,30H THE FIELD HAS BEEN SCALED DOWN/)
24 CONTINUE

```

IDENT JITTER

```

*IDENT JITTER
*DELETE GDL.20
X ICUT,MLT,IDK,IJTH,
X ICNT24,ICNT25,ICNT26,
X ITM,ICFK,NCT
*DELETE GDL.26
XDSMM(20),RMV(20),PHIA(20),RCURVE(4),OSP(4),TLT(4),ICAVZ(20)
*DELETE GDL.260
DO 173 IZER0=1,20
*DELETE GDL.314
DO 177 IZER0=1,17
*INSERT SQ077CY1.165
C = 23 JITTERS THE BEAM AN ANGLE ANGJIT
C = 24 DUMMY - LINE 240 IS TEMPORARILY STORED IN JITTER IFLOW.
C = 25 DUMMY - LINE 250 IS TEMPORARILY STORED IN JITTER IFLOW.
C = 26 DUMMY - LINE 260 IS TEMPORARILY STORED IN JITTER IFLOW.
*INSERT CORR2.7
DATA IJTR,JITANG,JITDIS /0.0,0.0,0.0/
*DELETE GDL.295,SQ077CY1.167
C / 16/ 17/ 18/ 19/ 20/ 21/ 22/ 23/ 24/ 25/ 26/
X.160,170,180,190,200,210,230,240,250,260),IFLOW
*DELETE GDL.325,SQ077CY1.168
C / 16/ 17/ 18/ 19/ 20/ 21/ 22/ 23/ 24/ 25/ 26/
X.160,170,180,190,200,210,230,240,250,260),IFLOW
*INSERT GDL.243
C
C NAMELIST /JITTER/ JITANG,JITDIS
C
C JITANG = THE ANGLE OF JITTER (IN MICRORADIANS)
C JITDIS = THE DISTANCE PROPAGATED SUCH THAT THE JITTER
C SIGMA IS JITANG*JITDIS*1.E-6

```

```

C
*DELETE GDL.327
C .....
C
230 IJTR = IJTR+1
IF(.NOT.INIT) GO TO 231
READ(IN,JITTER)
AHC(6,IJTR,1) = JITANG*1.E-6
ARC(6,IJTR,2) = JITDIS
231 SIGXY=AHC(6,IJTR,1)*ABC(6,IJTR,2)
WRITE (6,1836) AHC(6,IJTR,1),SIGXY
1836 FORMAT (45H **** BEAM JITTER MODEL CALLED ****,STD DEVIA.
X23HTION ANGLE (RADIAN) = ,G12.5,8X,*, STD DEVIATION (SIG4Y) =*,
XG12.5)
DO 233 IJ=1,NOR
233 US(IJ) = CUR(2*IJ-1)**2 + CUR(2*IJ)**2
DX = X(2) - X(1)
CALL JITHRG(DX,SIGXY)
DO 235 IJ=1,NOR
IJIJ = 2*IJ
CUR(IJIJ-1) = SQRT(US(IJ))
235 CUR(IJIJ) = 0.0
240 CONTINUE
250 CONTINUE

260 CONTINUE
IGAL = 1
GO TO 644
*INSERT GDL.32
EQUIVALENCE (US(1),CFIL(1)),(CUR(1),CU(1))
DIMENSION US(16384),CUR(32768)
*INSERT GDL.15
X = US,CUR
REAL JITANG,JITDIS
*INSERT CYCLE 9.67
DO 134 I=1,NPTS
134 X(I) = XSAVE(I)
*DELETE QUAL.2
SUBROUTINE QUAL (IPHASE,ISAVE,IPLI,TITLE,PH,ANS,PH,HF,SIGANG)
*INSERT QUAL.11
DATA PI/3.14159266/
*DELETE QUAL.107
63 IF(SIGANG-1.E-4)70,70,64
66 SIGXY=F*SIGANG
WRITE (6,1836) SIGANG
1836 FORMAT (45H **** BEAM JITTER MODEL CALLED ****,STD DEVIA.
X14HTION ANGLE = ,G20.5)
CALL JITHRG(DXSAVE,SIGXY)
UMAX=0.
DO 64 J=1,NPTS
JI=(J-1)*NPTS
DO 66 I=1,NPTS
I2=I+JI
IF (UMAX.GE.US(I2))GO TO 66
UMAX=US(I2)
XPEAK=X(I)
YPEAK=Y(J)
I2I2=2*I2
CUR(I2I2-1) = SQRT(US(I2))
CUR(I2I2) = 0.0
66 CONTINUE
70 UMX=UMAX/1000.
*INSERT MAIN.700
SUBROUTINE JITHRG(DX,SIGXY)
C THIS SUBROUTINE MODIFIES THE FAR FIELD INTENSITY DISTRIBUTION

```

```

C      MODEL THE EFFECTS OF BEAM JITTER. THE JITTER IS ASSUMED TO
C      GAUSSIAN. SINCE THE RESULTING INTENSITY IS THE CONVOLUTION OF THE
C      UNDISTURBED INTENSITY WITH THE GAUSSIAN, THE OPERATION IS PERFORMED
C      BY THE FFT ON EACH FUNCTION ALONE, MULTIPLYING THE RESULTS,
C      PERFORMING THE INVERSE FFT. JVF.6/24/76.
LEVEL 2,00,01
COMPLEX CU,CI
COMMON /MELT/CH(143H4),US(33024),X(12H),WL,NPTS,NPY,DRX,DHY
COMMON /CG/CI(17100)
DIMENSION NND(2)
DATA WT /3.141593/
NPTSQ = NPTS * NPTS
PPW = 0.0
PPW = 0.0
C) 10 M=1,NPTSQ

      CI(M) = CMPLX (US(M),0.0)
10 PPW = PPW+US(M)
   NND(1)=NPTS
   NND(2)=NPTS
   NAR = 2*NPTSQ
   NPO2=NPTS/2
   CALL FOURT (CI,NAR,NND,1)
   SIGEXP = 2.0*(SIGXY * PI)**2
   SIDE=(X(NPTS)-X(1))/2. * DX/2.
   DF=.5/SIDE
   DO 20 J=1,NPTS
     YSQ=((J-1)*DF)**2
     IF (J.GT.NPO2) YSQ=((J-NPTS-1)*DF)**2
     JK=(J-1)*NPTS
     DO 20 I=1,NPTS
       XSQ=((I-1)*DF)**2
       IF (I.GT.NPO2) XSQ=((I-NPTS-1)*DF)**2
       K=I+JK
20 CI(K)=CI(K)*EXP(-SIGEXP*(XSQ+YSQ))
   CALL FOURT (CI,NAR,NND,-1)
   DO 30 KK=1,NPTSQ
     US(KK)=CABS(CI(KK))/NPTSQ
30 PPWN = PPWN+US(KK)
   PWRFAC = PPW/PPWN
   DO 40 MM=1,NPTSQ
40 US(MM) = US(MM)*PWRFAC
   WRITE(6,100) PWRFAC
100 FORMAT(/5X,*,THE POWER HAS BEEN SCALED BY A FACTOR OF*,G12.5,
      X *,IN SUBROUTINE JITHG,*/ )
   RETURN
   END
*DELETE MAIN,60
      NAMELIST/QLOT/TITLE,IULT,OH,ISAV,IPHASE,RBH,RF,SIGANG
*DELETE MAIN,230
210 CALL QVAL (IPHASE,ISAV,IULT,TITLE,RBH,AS,OB,RF,SIGANG)
*INSERT MAIN,22
      DATA SIGANG /0.0/

```

APPENDIX B
SOQ USERS GUIDE UPDATES
JUNE 1978 TO JANUARY 1979

INTRODUCTION

This appendix documents those changes made to the initial SOQ code between June 1978 and January 1979. The changes incorporated in the code are those that have become generally useful for the physical optics simulation problems which have been solved using the SOQ code. The users guide updates are also prepared to clarify and correct the initial description of the SOQ code as documented and delivered to AFWL on 1 March 1978, in the Preliminary SOQ Users Guide. This document supercedes previous written material on SOQ code documentation. The organization of the SOQ Users Guide Updates is as follows:

Section BI

New Subroutines

1. Subroutine ZERN
2. Subroutine CPUTIM
3. Subroutine LISTER

Section BII

Code Changes/Correction

BI. NEW SUBROUTINES

1. SUBROUTINE ZERN

Zernike polynomial terms give the SOQ code the ability to model mirrors with arbitrary surfaces. This subroutine also provides the determination of sensitivity of a given system to the level of these Zernike terms.

a. Relevant formalism -- The Zernike Polynomials are an orthogonal set of polynomials used to describe phase front aberrations. The low order terms of this set correspond to the low order Gauss-Seidel aberrations, such as piston, tilt, defocus, astigmatism, coma, and clover. A list of these polynomials, $Z(k)$, is given in Table B-1.

TABLE B1. ZERNIKE POLYNOMIALS

k	Z_k	k	Z_k
1	1.0	13	$(4R^4 - 3R^2) \sin 2\theta$
2	$R \cos \theta$	14	$R^4 \cos 4\theta$
3	$R \sin \theta$	15	$R^4 \sin 4\theta$
4	$2R^2 - 1$	16	$(10R^5 - 12R^3 + 3R) \cos \theta$
5	$R^2 \cos 2\theta$	17	$(10R^5 - 12R^3 + 3R) \sin \theta$
6	$R^2 \sin 2\theta$	18	$(5R^5 - 4R^3) \cos 3\theta$
7	$(3R^3 - 2R) \cos \theta$	19	$(5R^5 - 4R^3) \sin 3\theta$
8	$(3R^3 - 2R) \sin \theta$	20	$R^5 \cos 5\theta$
9	$R^3 \cos 3\theta$	21	$R^5 \sin 5\theta$
10	$R^3 \sin 3\theta$	22	$20R^6 - 30R^4 + 12R^2 - 1$
11	$6R^4 - 6R^2 + 1$	23	$70R^8 - 140R^6 + 560R^4 - 210R^2$
12	$(4R^4 - 3R^2) \cos 2\theta$	24	$252R^{10} - 630R^8 + 560R^6 - 210R^4$
			$+ 30R^2 - 1$

The phase applied is

$$\Delta \phi = \sum_{k=1}^{24} 2\pi P_k Z_k(R, \theta)$$

$$= \Delta \phi(1, \theta)$$

$$\frac{r}{R_0} = R < 1$$

$$R > 1$$

(B1)

If the Zernike radius R_0 is specified to be zero, it is a flag to set the phase identically equal to zero.

b. Fortran formalism -- Subroutine ZERN is called by GDL with IFLOW = 24. Namelist ZERNS contains the Zernike radius R_0 as well as the coefficients of the Zernike polynomials to be applied $P(I)$ $I = (1, 24)$.

Due to excessive use of the FRINGE program, one can also input fringe coefficients (PFRNG(1)), corresponding to the 24 Zernike polynomials to be applied. The PFRNG coefficients are converted to P coefficients in subroutine GDL.

NAMELIST /ZERNS/	RO, P, PFRNG
Argument List	RO, P
Commons	/MELT/
Externals	None

IDENT ZRNIKE computer printouts follow.

1 15 17 7-14154

• 1111 11 11111111

40. 24

[illegible]

```

      P(17) = DEFN1(14)
      P(18) = DEFN1(14)
      P(19) = DEFN1(13)
      P(20) = DEFN1(25)
      P(21) = DEFN1(26)
      P(22) = DEFN1(15)
      P(23) = DEFN1(24)
      P(24) = DEFN1(15)
      IFIRST = 0
      DO 245 K=20,23
245  IF (DEFN1(K), F,0.) IFIRST = 1
      IF 245 K=7,14
245  IF (DEFN1(K), F,0.) IFIRST = 1

      IF (IFIRST.F,1) WRITE(4,247)
247  FORMAT(15X,'FRINGE COEFFICIENTS OF ORDER 20 THROUGH 23',
     * AND 27 THROUGH 36 ARE IGNORED')
247  DO 242 I=1,24
242  PZSAVE(1,I/ERN) = P(I)
      PZSAVE(25,I/ERN) = 0
244  CALL ZERN(PZSAVE(25,I/ERN),PZSAVE(1,I/ERN))
      IGNU = 1
      GO TO 444
*DELETE 30L,27
      DIMENSION IPTS(50),PZSAVE(25,10),P(24),IPLT(20),
     * PERN(15),PZABS(9),PZINDS(9),OTILL(9,20)
*INSERT 30L,33
      DATA P,PERN/24*0.,35*0./, 50 / 5. /
*INSERT 30L,243
C
      NAMELIST /ZERN/ R0,P,PERN
C
      R0 = ZERN[AK NORMALIZATION RADIUS.
      P = ARRAY OF ZERNIKE COEFFICIENTS TO BE APPLIED.
      PERN = ARRAY OF FRINGE ZERNIKE COEFFICIENTS TO BE APPLIED.
TILT
*INSERT 100L,745
      SUBROUTINE ZERN(R0,P)
      LEVEL 2,COM
      COMMON /ZERN/ CUP(12/64),CFL(16/12),X(128),WL,NPTS,NPY,DHX,DYX
      COMPLEX CFTI
      DIMENSION P(24)
      IF (R0.EQ.0.) GO TO 70
      DO 100 IY=1,100
      J1 = (IY-1)*NPTS
      YSU = X(IY)*R0
      DO 100 IX=1,100
      XSU = X(IX)*R0
      INDX = IX + J1
      R = SQRT(XSU*YSU)
      THET = ATAN2(X(IY),X(IX))
      R = AMIN1(R/R0,1.)
      CT = COS(THET)
      C2T = COS(2.*THET)
      C4T = COS(4.*THET)
      C6T = COS(6.*THET)
      C8T = COS(8.*THET)
      ST = SIN(THET)
      S2T = SIN(2.*THET)
      S4T = SIN(4.*THET)
      S6T = SIN(6.*THET)
      S8T = SIN(8.*THET)
      W2 = W0*2
      W4 = W0*4
      W6 = W0*6
      W8 = W0*8

```

```

      W4 = W2*W4
      W10 = W2*W4

      DEL = W(1) + W(2)*W2*CT + W(3)*W4*ST
      A = W(4)*W(2)*W2-1.)

      C = W(5)*W2*CT + W(6)*W2*ST
      D = W(7)*(4.*W2-2.*W)*CT + W(8)*(3.*W3-2.*W)*ST
      E = W(9)*W4*CT + W(10)*W2*ST
      F = W(11)*(6.*W2-4.*W2+1.)
      G = W(12)*(4.*W4-3.*W2)*CT + W(13)*(4.*W4-3.*W2)*ST
      H = W(14)*W4*CT + W(15)*W4*ST
      I = W(16)*(13.*W4-12.*W3+3.*W)*CT
      J = W(17)*(13.*W4-12.*W3+3.*W)*ST
      K = W(18)*(5.*W4-4.*W2)*CT + W(19)*(5.*W4-4.*W3)*ST
      L = W(20)*W4*CT + W(21)*W4*ST
      M = W(22)*(20.*W4-30.*W4+12.*W2-1.)
      N = W(24)*(70.*W4-140.*W4+90.*W4-20.*W2+1.)
      O = W(26)*(232.*W10-641.*W4+760.*W4-210.*W4+30.*W2-1.)

      INO2 = 100*W2
      DEL = DEL*2.*1.15197654
      COS1 = COS(DEL)
      SIN1 = SIN(DEL)
      CWS = COS(INO2-1)
      CUR1(W2-1) = CWS*COS1 = CUR(INO2)*SIN1
      CUR1(INO2) = CWS*SIN1 = CUR(INO2)*COS1
      WRITE (N,203) W2-2
200 FORMAT (102F9.4) PHASE CORRECTION APPLIED WITH NORMALIZATION*
      A = RADIUS OF 0.0154 / COEFFICIENTS USED W(1)-W(24)*.
      B = ARE CONSISTENT WITH THE PHASE DUE TO THE NTH TERM BEING*//
      C 203.24= W(10) = 203*W(10)*7(N)//
      D = Z(N) = W(10)*100*W(10)*W(10) W(10) NORMALIZED TO 1. AT R=1.0//
      E (1.5020.5)
      RETURN
201 END = NPT5*W2
      DO 40 I=1,100
      II=I+1
      LIM=II-1
      CUR(LIM) = 50*W(CUR(II)*2+CUR(LIM)*2)
40 CUR(II) = 1.0
      WRITE (5,300)
300 FORMAT (11F9.4) PHASE HAS BEEN SET TO ZERO IN SUBROUTINE ZERN*//
      RETURN
      END

```

2. SUBROUTINE CPUTIM

Subroutine CPUTIM has been activated for the CDC computer to print out the amount of CPU seconds used by the kinetics package, which is driven by Subroutine REGAIN. On the Cyber 176 a system routine

A = Second (B)

returns the CP time since start of job, in seconds, to both A and B.

FORTTRAN:

Argument List:

IT = 100* time since start of program

Commons None

Externals None

IDENT CPUTIM computer printout follows.

IDENT CPUTIM

```
*IDENT CPUTIM
CPUTIM
  *DELETE DUMMYS.20
    SUBROUTINE CPUTIM(IT)
      IT = 100*SEC/100(IT)
  *GAIN
  *DELETE *GAIN.03
    DELT = (ITEN-ISWT)/100.
```

3. SUBROUTINE LISTER

Subroutine LISTER was activated so that the output of the resonator design program RESDES or an arbitrary file may be read internally and reprinted in the output of the SOQ code. LISTER reads an 80-column file, designated as Tape K, and reproduces it in the SOQ-designated system output file with pagination defined the same as on Tape K.

FORTTRAN:

LISTER is called anytime IRSDS, is nonzero in namelist START.

Argument List:

K (= IRSDS from START)
= tape number of the file to be replicated

Commons: None

Externals: None

IDENT LISTER computer printout follows.

IDENT LISTER

```
*IDENT LISTER
LISTER
  *DELETE DUMMYS.23,DUMMYS.25
    SUBROUTINE LISTER(K)
  C *** THIS ROUTINE COPIES TAPE K TO OUTPUT.
    DIMENSION C(20)
    REWIND K
  C
    1 HEAD(K,5) IC1=C
      IF (EOK(K).NE.0.0) GO TO 15
    5 FORMAT(11,20A4)
      IF (IC1.EQ.1) *WRITE(6,35)
      *WRITE(6,10) C
      10 FORMAT(11X,20A4)
  C READ THE NEXT CARD
      GO TO 1
```

```

C
15 REWIND K
WRITE (6,35)
35 FORMAT(1H1)
RETURN
END

MAIN
*INSERT MAIN.155
IF (IRSOS.NE.0) CALL LISTER(IRSOS)
IF (NWL.LE.0.) REWIND 50
IRSOS = 0

```

BII. CODE CHANGES/CORRECTIONS

The code modifications and corrections included in the code are described below by their update file name. The reason for the change, the structure, and the listing are included below:

1. *ID SOQMAP

This update provides a cross-reference map to the SOQ79128 code. The first section lists each routine in the order of appearance in the SOQ code with its commons and externals. Also given is a list of all routines that call it. The second section lists every common block in the SOQ code with the subroutines possessing that common block.

IDENT SOQMAP

```

*IDENT SOQMAP
MAIN
*INSERT MAIN.23
C
C FOLLOWING IS A ROAD MAP FOR THE SOQ CODE CROSS-REFERENCING
C COMMONS AND EXTERNALS.
C
C NOTE: COMMONS AT LEVEL 2 ARE FOLLOWED BY "(2)".
C
C ROUTINE      COMMON      EXTERNAL      CALLED BY
C -----
C SOQ          EST (2)      CNSTRN
C              MFLT (2)     DAVIDN
C              PLTSTG       GOL
C              INITL        ISOS
C              GLAD          LISTER
C              SVTYP         LISTM
C              NAMES        MPAR
C              PRETYP        QUAL
C
C JITRNG       MFLT (2).    FDMHT      GOL
C              CG (2)       QUAL
C
C ISQ          -            -            -
C ERFC         -            ERF          GATNXY
C
C
C
C

```

ISOCAM	-	-	REGAIN
ISOS	-	-	SOC
PRETYO	-	-	SOC
THREED	-	-	NEAR
VIND	-	-	REGAIN
CONTIN	-	-	FILMS
			GAINXY
			GOL
			REGAIN
LISTER	-	-	SOC
LISTR	-	-	SOC
DATE	-	-	NEAR
HLOCK	-	-	NEAR
AEROW	MELT (2)	-	GOL
HANDU	-	-	-
APHTR	MELT (2)	-	FIELDS
	WAY	-	GOL
			MIRROR
			GOL
AXION	MELT (2)	INTERP	
		SPTAN	
HUMIT	CAV2 (2)	-	REGAIN
CAVITY	GEACTH	DENSY	GOL
	WAY	GAINXY	
	CAV2 (2)	INTERP	
	MELT (2)	OUTPUT	
	CCG (2)	STEP	
	GLAD		
	CAVX (2)		
	APPROX		
	SEMCV2		
	STPWL		
CEHAR	-	-	QUAL
CUSTOM	-	-	SOC
CONF	MELT (2)	-	NEAR
CAVITY	-	-	SOC
DENSY	MELT (2)	LINTERP	CAVITY
	DENSY	RUSO	
	GLAD	RUSNA	
	SENDER		
FIELDS	MELT (2)	APHTR	GOL
FOUR	-	-	JITHRG
			RSTEP
			STEP
			REGAIN
FILMS	CAV2 (2)	CONTIN	
	GLAD		
GAINXY	STANT	CONTIN	CAVITY
	GEACTH	ENEC	REGAIN
	PHOT	KINET	
	MOLES	MIX	
	ENERG		
	RATE		
	FACTOR		
	CAV2 (2)		
	GLAD		
GOL	MELT (2)	AEROW	SOC
	APPROX	APHTR	
	WAY	AXION	
	ZTP	CAVITY	
	CAV	CONTIN	

	INITI	FIELDS	
	SEGGOL	INTERP	
	STPGWL	IMLIT	
	SVTYN	JITHRG	
	HAKES	MIRROW	
		POWR	
		REGAIN	
		RGRO	
		RSTED	
		SLIVER	
		SPIDER	
		STEP	
		TALOOM	
		THERML	
		ZERN	
INTERP	-	-	AXION
			CAVITY
			GOL
IPLOT	MELT (2)	OUTPUT	GOL
	WAY	OUTPUT	QUAL
KINET	PLTSTG		
	DOORT	EREC	GAINXY
	START	MIX	
	MOLES		
	ENERG		
	RATE		
	RACIER		
	GEACTH		
MIRROW	MELT (2)	APRTR	GOL
	MORROW		
	WAY		
MIX	DOORT	-	GAINXY
	MOLES		KINET
	RATE		
MORROW	MELT (2)	SPTAN	-
	WAY		
NEAN	MELT (2)	COHEFF	SOO
	TIME	HATE	
	VIEW	HCLUCK	
		THREED	
OUTPUT	WAY	-	IPLOT
OUTPUT	WAY	-	CAVITY
	PLTSTG		IPLOT
			TALOOM
			THERML
PLTOT	-	POWROW	QUAL
POWR	-	-	GOL
POWROW	-	-	PLTOT
			QUAL
QUAL	MELT (2)	CENHAR	SOO
	STPGWL	JITHRG	
		IPLOT	
		PLTOT	
		POWROW	
		STEP	
		TILT	
REGAIN	MELT (2)	ALUMIT	GOL

C		CCG (2)	CRUTIM	
C		CAV2 (2)	FUHS	
C		GLAD	GAINXY	
C			ISOCAY	
C			SIMPGG	
C			VINO	
C	ROGR	MFLT (2)	-	GOL
C	ROSN	LENSY	-	DENSY
C	LINTERP	LENSY	-	DENSY
C	ROSNA	MFLT (2)	-	DENSY
C	RSTEP	WAY	FOURT	GOL
C		MFLT (2)		
C	SIMPGG	CAV2 (2)	-	REGAIN
C		GFACTR		
C	SLIVER	MFLT (2)	-	GOL
C	SPIDER	MFLT (2)	-	GOL
C	SPTAN	-	-	AXION
C				MODER
C	STEP	WAY	FOURT	CAVITY
C		MFLT (2)	TILT	GOL
C		STPLCM (2)		DUAL
C		STRWVL		TALOOM
C	TALOOM	MFLT (2)	OUTPUT	GOL
C		WAY	STEP	
C	THERML	MFLT (2)	ERFC	GOL
C		WAY	OUTPUT	
C	TILT	MFLT (2)	-	DUAL
C				STEP
C	ERF	-	-	ERFC
C	ZERN	MFLT (2)	-	GOL
C	-----			
C	ROUTINES			
C	-----			
C	HARS	SQO.GOL		
C	CAVY (2)	CAVITY		
C	CAV2 (2)	ALUMIT.CAVITY.FUHS.GAINXY.REGAIN.SIMPGG		
C	CCG (2)	CAVITY.REGAIN		
C	CG (2)	JITRNG		
C	ENERG	GAINXY.KINET		
C	FACTER	GAINXY.KINET		
C	FST (2)	SQO		
C	GFACTR	CAVITY.GAINXY.KINET.SIMPGG		
C	GLAD	SQO.CAVITY.DENSY.FUHS.GAINXY.REGAIN		
C	INITL	SQO.GOL		
C	LENSY	DENSY.ROSN.LINTERP		
C	MFLT (2)	SQO.JITRNG.AFROW.APRTH.AXION.CAVITY.CONFFE.		
C		DENSY.FIELDUS.GOL.IPLOT.AIPROW.MODER.NEAP.DUAL.		
C		REGAIN.ROGR.ROSNA.RSTEP.SLIVER.SPIDER.STEP.		
C		TALOOM.THERML.TILT.ZERN		
C	MOLES	GAINXY.KINET.MIX		
C	APPROX	CAVITY.GOL.MIPROW		
C	PLTSIG	SQO.IPLOT.OUTPUT		
C	PROPT	GAINXY.KINET.MIX		
C	JAZ	GOL		

```

C      RATE          GAINXY,KINET,MIX
C      SEGCVP        CAVITY
C      SEGENEN       DENSITY
C      SEGGOL        GOL
C      START         GAINXY,KINET
C      STPLCM (2)    STEP
C      STPJWL        CAVITY,GOL,QUAL,STEP
C      SVTYM         SOQ,GOL
C      TIME          NEAR
C      VIEW          NEAR
C      WAY           APRTR,CAVITY,GOL,[PLOT,MIRROR,MODER,OUTPUT,
C                   OUTPUT,RSTEP,STEP,THLOOM,THERML
C      ZTP           GOL
C      -----
C
C
C

```

2. *ID ABCMAP

Current allocations of the ABC (I, J, K) array are presented here for ease in future updating.

IDENT ABCMAP

*IDENT ABCMAP

GOL

*INSERT GOL.245

C

C FOLLOWING IS A SUMMARY OF THE ABC ARRAY LOCATIONS USED
C IN GOL. ABC IS DIMENSIONED TO (14,20,9).

C

C	ABC(1,1,1)	THROUGH	ABC(4,1,1)	:	IFLOW=6, CUTOUT
C	ABC(1,2,1)	THROUGH	ABC(2,2,1)	:	DRX, DRY IN SOQ
C	ABC(1,IMIR,2)	THROUGH	ABC(14,IMIR,2)	:	IFLOW=2, MIRROR
C	ABC(10,IMIR,4)	THROUGH	ABC(13,IMIR,4)	:	IFLOW=2, MIRROR
C	ABC(1,ISTEP,3)	THROUGH	ABC(4,ISTEP,3)	:	IFLOW=3, PROP
C	ABC(1,AP,4)	THROUGH	ABC(4,AP,4)	:	IFLOW=4, APRTR
C	ABC(1,IOK,5)	THROUGH	ABC(14,IOK,5)	:	IFLOW=5, THLOOM
C	ABC(1,IJTR,6)	THROUGH	ABC(2,IJTR,6)	:	IFLOW=23, JITTER
C	ABC(10,IWFA,6)			:	IFLOW=15, REGHID
C	ABC(1,ITHHML,7)	THROUGH	ABC(4,ITHHML,7)	:	IFLOW=17, THERML
C	ABC(1,IHSTEP,8)	THROUGH	ABC(4,IHSTEP,8)	:	IFLOW=20, RSTEP
C	ABC(1,MLT,9)	THROUGH	ABC(2,MLT,9)	:	IFLOW=12, MULT

C

C

C

3. *ID PLTFIX

Ident PLTFIX modifies the printer-plotting package in the SOQ code.
This new plot package:

- Prints DCALC, IMAX, DCALC FLUX along with the location of the center of the beam (DRX, DRY) and the bottom of every iso-intensity plot

- b. Prints a blank for every value of intensity less than 0.01
*UMAX (UMAX is maximum intensity) and puts a border around
the outside in column 1 of NPTS and row 1 to NPY
- c. Allows for selective plotting, based on the new namelist
parameter KPLOT in namelists PLOT and QLOT.

IDENT PLOTIX

*IDENT PLOTIX

IPLOT

```
*[INSERT LROP1.66
    DCF = 0.
*[INSERT LROP1.66
    DCF = DCF + US(I)
*[INSERT LROP1.67
    DCF = DCF*(X(2)-X(1))**2*FLOAT(NPTS/NPY)/1000.
*[DELETE LROP1.86,LROP1.87
    34 WRITE(6,6) A(I),UMAX,DCF,UMAX,DY
    6 FORMAT(12H    DCONF = ,G11.5,4X,7HIMAX = ,G11.5,4X,6HDCALC ,
    X 7HFLUX = ,G11.5//24X)
*[DELETE LROP1.96
    1500 IF(XAXIS) WRITE(6,746)
```

GOL

```
*[DELETE GOL.154
    NAMELIST /PLOT/ KPLOT,TITLE,RADPLOT
C          KPLOT = 4*DCF, WHERE A,M,C,D,AND E ARE 0 OR 1.
C          A = 1      RADIAL PLOTS
C          M = 1      ISOINTENSITY PLOTS
C          C = 1      X - AXIS PLOTS
C          D = 1      DIAGONAL PLOTS
C          E = 1      Y - AXIS PLOTS
*[INSERT GOL.503
    KPLOT = 0
*[INSERT GOL.505
    IF(RADPLOT.NE.0..AND.KPLOT.EQ.0) KPLOT=1111
    IF(RADPLOT.EQ.0..AND.KPLOT.EQ.0) KPLOT=1111
    IIPLOT(IPTT) = KPLOT
*[DELETE GOL.510,GOL.511
    KPLOT = IIPLOT(IPTT)
    CALL IPLOT(KPLOT)
```

IPLOT

```
*[INSERT LROP1.30
    DIMENSION X(14),XII(124)
*[INSERT LROP1.35
    DATA BLANK,001 /14 ,14, /
    DATA 9 /14 ,140,141,142,143,144,145,146,147,148,149,140,141, /
*[DELETE LROP1.75,LROP1.74
    U1 = US(12)/UMAX
    IK = [0,001] * 2
    IF(U1.LT..01) IK=1
    IF(I.EQ.1.(0,1.E0,NPY) IK=14
    IF(J.EQ.1.(0,1.E0,NPTS) IK=13
    2 XII(I) = X(IK)
    4 WRITE(6,7) (XII(I),I=1,NPY)
    3 FORMAT(1X,124A1)
```

```

*DELETE L2001,M3,L2001,45
  U1 = US(I2)/UMAX
  IK = 10.001 + 2
  IF(U1.LT..01) IK=1
  IF(I.EQ.1.OR.I.EQ.400) IK=14
  IF(I.EQ.1.OR.I.EQ.4005) IK=13
  12 X(I) = X(IK)
  14 WRITE(6,13) X(I),X(I(I),I=1,400)
  13 FORMAT(1X,F10.2,2(1.64A))

```

4. *ID ADDPRNT

This section of updates was included to add information on intermediate printout to CAVITY, STEP, GDL, and TILT:

- a. CAVITY - The incoming and outgoing total flux at each gain/phase section is now printed.
- b. STEP - At the beginning of STEP, current values for DRX, DRY, RAPTR, NREG, and WNOW are printed, and the incoming flux calculated. At the end of STEP, modified values of DRX, DRY, NREG, and WNOW are printed along with the percent flux lost during the propagation step. This last parameter (percent flux lost) indicates how much of the beam has been propagated out of the calculation mesh and, therefore, lost by windowing in S-space and K-space (Fourier Transform Space).
- c. GDL - At the end of any IFLOW the code now prints out total DCALC FLUX, DCALC, and the location and magnitude of IMAX.
- d. TILT - Subroutine TILT now prints out the mirror radius of curvature necessary to remove the beam radius of curvature found by TILT.

IDENT ADDPRNT

*IDENT ADDPRNT

ID ADDPRNT ADDS MORE INFORMATION TO OUTPUT FROM SUBROUTINES CAVITY, STEP AND TILT.

CAVITY

```

*INSERT CAVITY,20%
  POWA = 0.
  POWB = 0.

```

```

*DELETE CAVITY.244
  XMAX = 2*MX
  XINT = (CUR(XMAX)**2 + CUR(XMAX-1)**2)*XFACT
  POWH = POWH + XINT
  A1 US(MX) = XINT
  POWH = (POWH*(X(2)-X(1))**2*FLOAT(NPTS/NPY))/1000.
*DELETE CAVITY.324
  JYJY = 2*JY
  XINT = (CUR(JYJY)**2 + CUR(JYJY-1)**2)*XFACT
  POWA = POWA + XINT
  A4 US(JY) = XINT + US(JY)
  POWA = (POWA*(X(2)-X(1))**2*FLOAT(NPTS/NPY))/1000.
  WRITE(6,62) JNS,NCAYN,POWH,POWA
  A2 FORMAT(/3X,14HGA IN/PHASE SEGMENT.12.17H IN CAVITY NUMBER.12.
  X24H HAS BEEN APPLIED. FLUX IN =.G14.7.134. FLUX OUT =.G14.7/)
STEP
*INSERT STEP.14
  DATA NREG,WNOW /0.1,0/
  IF(IPRNT.NF.0) WRITE(6,1000) DXREAL,DYREAL,RAPTH,NREG,WNOW
1000 FORMAT(5X,10HSTEP IS SUBROUTINE STEP. CURRENT PROPAGATION*.
  X * PARAMETERS:*.
  X /X1,0HXX =.G12.4.
  X /X1,0HY =.G12.4.
  X /X1,0HAPTH =.G12.4.
  X /X1,0NREG =.19.
  X /X1,0WNOW =.G12.4.
  POWH = 0.
  POWA = 0.
  NOK = NPTS*NPY
  DO 400 I=1,NOK
    II = 2*I
    POWH = POWH + CUR(II-1)**2 + CUR(II)**2
    POWH = POWH*(X(2)-X(1))**2*FLOAT(NPTS/NPY)/1000.
    IF(NREG.EQ.1.0H,NREG.EQ.2) POWH = POWH/WNOW**2
*INSERT STEP.234
  NOK = NPTS*NPY
  DO 401 I=1,NOK
    II = 2*I
    POWA = POWA + CUR(II-1)**2 + CUR(II)**2
    POWA = POWA*(X(2)-X(1))**2*FLOAT(NPTS/NPY)/1000.
    IF(NREG.EQ.1.0H,NREG.EQ.2) POWA = POWA/WNOW**2
    DELP = (POWH-POWA)/POWH*100.
    IF((ITH.EQ.0.0H,NREG.EQ.0).AND).IPRNT.NF.0) WRITE(6,3000)
    X DXREAL,DYREAL,NREG,WNOW,DELP
1000 FORMAT(14X,10HEXITING SUBROUTINE STEP. CURRENT PROPAGATION*.
  X * PARAMETERS:*.
  X /X1,0HXX =.G12.4.
  X /X1,0HY =.G12.4.
  X /X1,0NREG =.19.
  X /X1,0WNOW =.G12.4.
  X /X1,0PERCENT FLUX LOST =.G12.4.
*DELETE STEP.191
  16 FORMAT(14X,14H STREAM INTENSITY =.G12.5)
*INSERT STEP.254
  WNOW = 1.0
  IF(IPRNT.NF.0) WRITE(6,3000) DXREAL,DYREAL,NREG,WNOW,DELP
*INSERT STEP.264
  WNOW = 1.0

```

GUL

```

*DELETE GDL.H47,GDL.H48
  UMAX = 0.0
  XMAX = X(1)
  YMAX = X(1)
  DO 74 J=1,NPY
    J1 = (J-1)*NPTS
    DO 74 I=1,NPTS
      IZ = I+J1
      XYINT = CU(I/1)*CONJG(CU(I/1))
      IF(UMAX.GT.XYINT) GO TO 74
      UMAX = XYINT
      XMAX = X(I)
      YMAX = X(J)
74   PPW = PPW*XYINT
      IF(NWEG.EQ.1.(NH.NWEG.EQ.2)) UMAX=UMAX/WNOW**2
      UMAXK = UMAX/1000.
      RADMAX = SQRT(XMAX**2+YMAX**2)
*DELETE GDL.H66
  IF(MSTEP.NF.1) WRITE(6,79) PPWK,DCALCP,UMAXK,XMAX,YMAX,RADMAX
*DELETE GDL.H70
  XUX = .G12.4/HX.12HDCALC = .FR.2/HX.12HIMAX = .G12.4.10X.
  X20HLOCATED AT (X,Y) = (.G12.4.1H..G12.4.1H).
  X /42X.9HAT RAD(US.G12.5)
*DELETE GDL.H71
  IF(MSTEP.EQ.1) WRITE(6,774) PPWK,DCALCP,UMAXK,XMAX,YMAX,RADMAX
*DELETE GDL.H73
  XUX = .G12.4/HX.12HDCALC = .FR.2/HX.12HIMAX = .G12.4.10X.
  X20HLOCATED AT (X,Y) = (.G12.4.1H..G12.4.1H).
  X /42X.9HAT RAD(US.G12.5)
*DELETE GDL.512,GDL.513
  IGVAL = 1
  GO TO 999

TILT
*DELETE CYCLE9.233
  TWOHDC = 2.*RADCUR
  IF(IPS.GF.2) WRITE(6,67) RADCUR,TWOHDC
*DELETE CYCLE9.235
  X 10X.32HMMASE FRONT CURVATURE = RADCUR = .G12.4.3H CM/
  X /10X.* NOTE - THIS CURVATURE CAN BE REMOVED WITH A MIRROR./
  X 10X.* USING RADC = .G12.4.31H= 2.*RADCUR AS DEFINED ABOVE - /
  X 10X.* NEGATIVE RADCUR IS A CONVERGING PHASE FRONT WHICH./
  X 10X.* CAN BE REMOVED WITH A CONVEX (NEGATIVE RADC) MIRROR./

  X 141)

```

5. *ID SCLPWR

Ident SCLPWR modifies the IFLOW = 12 section of GDL to allow for scaling of the beam to a specific power TRANS.

```
IDENT SCLPWR
```

```

*IDENT SCLPWR
  GDL
    *INSERT GDL.444
      XMAG = 1.
    *DELETE GDL.490,GDL.491
    C *** IF(TRANS.LE.1.0) THE FIELD IS SCALED BY SQRT(TRANS)/XMAG
    C *** IF(TRANS.GT.1.0) THE FIELD IS SCALED TO THE POWER "TRANS"

```

```

351 POLD = ARC(1.0,MLT.0)
AMAG = ARC(2.0,MLT.0)
TRANS = POLD
IF (TRANS.EF.1.0) GO TO 359
PNEW = 0.0
DO 356 IZ=1,400
356 PNEW = PNEW + CU(IZ)*CONJG(CU(IZ))
PNEW = PNEW*((X(2)-X(1))**2)*(NPTS/NPY)/1000.
IF (NNEG.EF.1.00.NNEG.EF.2) PNEW=PNEW/WNUW**2
359 IF (TRANS.EF.1.0) PNEW = AMAG**2
STRANS=SQRT(POLD/PNEW)
WRITE(6,352) ARC(1.0,MLT.0),ARC(2.0,MLT.0)
*INSERT G01.494
IF (TRANS.GF.1) IGAL=1

```

6. *ID TBLUM

Two errors in subroutine TBLUM are corrected by this ident. The following listing is self-explanatory.

```

IDENT TBLUM
*IDENT TBLUM
TBLUM
*DELETE TBLUM.42
IF (AXIAL.GF.0.0) WRITE(6,596) AXIAL
*DELETE TBLUM.46
C MT = (400.005*41*FA**2.*PI/(400*CP*T))**(.1/3.)

```

7. *ID REMSPH

Ident REMSPH allows the removal of defocus and/or tilt using a call to subroutine QUAL, and to continue with this optimized beam. This optimized field can be plotted and written to a local file specified by IWRITE.

IWRITE. FT.0 sets IW = IWRITE

IWRITE. LT.0 sets in IW = -IWRITE and returns to SOQ immediately.

If desired, the non-optimized field can be read in using ISAV = 1 in namelist QLOT.

```

IDENT REMSPH
*IDENT REMSPH
QUAL
*INSERT PIAL.51
IF (IW.EF.0.0) GO TO 60
IOW = IWRITE
I* = IWRITE

```

```

      IF (IWRITE.LT.0) IY = -IWRITE
      WRITE(IW) (C(I,X),IX=1,NOR),X,DRX,DY,NIT,SAVE
      REWIND IW
      WRITE(6,43) 11,IPHASE
49  FORMAT(11,5X,9C1) HAS BEEN WRITTEN ON UNIT*,I3,* FROM QUAL*,
      X * WITH IPHASE =*,I2//)
      IF (IWRITE.GE.0) GO TO 50
      IPNT = 1
      IF (KPIOT.GT.0) WRITE(6,3000) NETITL
      IF (KPIOT.GT.0) CALL IPLOT(KPIOT)
      IF (ISAVE.EQ.1) READ(7) (C(I,X),IX=1,NOR),X,DRX,DY
      REWIND 7
      RETURN
50  CONTINUE
MAIN
*DELETE MAIN.227
200  KPIOT=1000
      READ(5,9101)

```

8. *ID CHGNPT

Ident CHGNPT increases the flexibility of IFLOW = 6 in two ways:

- Reoverlap the beam, letting the code find the original DCALC by setting DIBEAM = 0
- Change the number of points in the beam, by interpolation, by specifying NEWNPT and NEWNPY. On a subsequent call to START, set NNPTS equal to the value of NEWNPT or NPTS will be reset to the previous value of NNPTS.

IDENT CHGNPT

```

*START CHGNPT
GOL
  *DELETE GOL.442
  150  IWEA=IWA + 1
      IF (.NOT. IN(I) GO TO 153
      READ(INWEA,10)
      ARC(10,IWEA,6) = 0.000
  153  NNEW = ARC(10,IWEA,6)
  *INSERT CYCLE 9.6
      NPTS = NEWNPT
      NPY = NEWNPY
      NNM = NPY*NPTS
  *INSERT CYCLE 9.7
      X = NEWNPT*NEWNPY*IMRSM
      C  NEWNPT AND NEWNPY ARE THE DESIRED NUMBER OF POINTS
      C  SAMPLING THE FIELD.
      C  IMRSM = 1/M FOR THE CVM. DEFAULTS TO 1/M = 1.
      C  IT IS USED TO RENORMALIZE THE HARK RESONATOR FEEDBACK FIELD.
  *INSERT GOL.457
      NEWNPT = 0
      NEWNPY = 0
      IMRSM = 1

```

```

*INSERT GDL.664
  ARC(6.1.1) = NEWNPT
  ARC(7.1.1) = NEWNPY
  ARC(8.1.1) = INIDSM
  IF (IDPEAM.EQ.0.) ARC(1.1.1) = (X(2)-X(1))*FLOAT(NPTS)
*DELETE GDL.667
  NEWNPT = ARC(6.1.1)+.01
  NEWNPY = ARC(7.1.1)+.01
  INIDSM = ARC(8.1.1)+.01
  IF (NEWNPT.EQ.0) NEWNPT=NPTS
  IF (NEWNPY.EQ.0) NEWNPY=NPY
  XDEL = 0.1*(X-NEWNPT)*2.
*DELETE GDL.669
  DO 62 IG=2,NEWNPT
*DELETE GDL.684
  DO 63 NY=1,NEWNPY
*DELETE GDL.686
  DO 64 NX=1,NEWNPT
*DELETE AMR2A.17
  NEWNOR = NEWNPT+NEWNPY
  WNDWSQ = 1.
  IF (NREG.EQ.1.OR.NREG.EQ.2) WNDWSQ=WNOW**2
  POWA = POWA/WNDWSQ
  POWB = POWB/WNDWSQ
  DO 623 IX=1,NEWNOR

```

9. *ID MISCFX

This ident corrects minor errors and adds two parameters to namelist START.

- a. *D GDL 384,385 This change in format compacts this part of the printout to 80 columns for 4 or fewer struts.
- b. *D GDL 884, 885 This change removes the S from column 1 so that output can be put on microfiche. It also corrects an error in the BARE updates so that the CU field is read in at the end of a converged iteration.
- c. *D CORR 1.23, 24 This change removes \$ from column 1 in the output.
- d. *D JITTER .83, 86 This change updates the indices in the ABC array which were defined originally in reverse order.
- e. *D BARE .11 This change corrects the size of the loop from MUT to NOB.

- f. *D Cycle 9. 119, 120 Previously for IPS = 2, the iteration counter KOUNT was not updated.
- g. *I Cycle 9.99 Focal = 1.E50 defaults the radius of curvature of the beam to "infinity."
- h. *I STEP .40 This change activates the IIPS ≠ 0 option in STEP. Setting DELZ = 0 allows the removal of tilt and/or calculated sphere without propagating the field.
- i. *D GDL .827 This statement was redundant.
- j. *D BARE .86 The parameter REGAIN allows the option of not calling REGAIN at the end of an iteration.

The parameter IFLGAP is included so that aperture loads are printed for all apertures in the optical train.

IDENT MISCEX

*IIF ID MISCEX

THIS ID CORRECTS MINOR ERRORS AND ADDS PARAMETERS TO NAMELIST START.

GDL

```
*DELETE GDL.184.GDL.385
      INO. OF STRUTS=.12.2X.12H X-Y CENTER=.610.4.1H.610.4./2X.
      2 13HWH DIAMETER=.610.4.3X.7HTHETAS=.6010.4)
*DELETE GDL.384.GDL.385
      ADD FORMAT(//1X.119(14) //33H          ITERATION IS CONVERGED .
      1 6H AFTER.14.16H ITERATIONS //1X.119(14) //)
      READ(9) (C(12).I2=1,NOR).X,ORX,ORY
      REFINO 9
```

LISTAD

```
*DELETE CORR1.21.CORR1.24
      =1X.4HCARD.15.10(141).10(142).10(143).10(144).10(145).
      =10(146).10(147).14H.//7H COLUMN.4X.4(10H1234567890).5X.
```

GDI

```
*DELETE JITTER.23.JITTER.26
      ARC(1.1JTR.5) = JITANG*1.E-6
      ARC(2.1JTR.6) = JITDIS
      231 SIGXY = ARC(1.1JTR.6)*ARC(2.1JTR.6)
      WRITE (6,1234) ARC(1.1JTR.6).SIGXY
```

CAVITY

```
*DELETE HAWF.11
      100 WRITE(7) (C(12).I2=1,NOR)
```

```

FILT
  *DELETE CYCLE4.119,CYCLE4.120
    26 COUNT = COUNT+1
    IF (IPS.EQ.2) GO TO 54
  *INSERT CYCLE4.40
    FOCAL = 1.550
STEP
  *INSERT STEP.40
    RADCOR = RADCOR
    IF (DFLZ.EQ.0.) RETURN
GOL
  *DELETE GOL.407
  *DELETE NAME.84
    IF (RGAIN) CALL REGAIN(NOT,ITER)
MAIN
  *INSERT MAIN.13
    LOGICAL RGAIN
  *INSERT NAME.4
  C   IRSDS IS THE TAPE NUMBER OF THE 80-COLUMN FILE TO BE COPIED TO
  C   OUTPUT BY LISTER. IF IRSDS=0, LISTER IS NOT CALLED.
  C   RGAIN = .FALSE. TURNS OFF THE CALL TO REGAIN IN IFLOW=7.
  *INSERT MAIN.154
    RGAIN = .TRUE.
    IRSDS = 0
    IFLGAP = 0
  *INSERT MAIN.221
    IFLGAP = 1
  *DELETE NAME.1
  C   IFRAME,PLOTS,RGAIN,IRSDS
  *INSERT MAIN.7
    COMMON /SVTYM/ RGAIN,IFLGAP
GOL
  *INSERT GOL.17
    COMMON /SVTYM/ RGAIN,IFLGAP
  *DELETE GOL.401
    IF (ICHTL.EQ.1.AND.IFLGAP.EQ.0) GO TO 998

```

10. *ID FXQUAL

The quality program has been updated to include more options and more printouts. See also *ID PROP and *ID RMVSPH for other additions to QUAL

- a. IPRNT. This parameter was added to suppress the additional STEP output (from *ID ADDPRNT) when STEP is called from subroutine QUAL. It was also added to namelist PROPGT for the same purpose.
- b. The output of the focal plane search was modified to print out more information.
- c. Additions to QLOT

RBB (New meaning)

IWRITE (see *ID RMVSPHP)

PROP (see *ID PROP)

IRYFF

KPLOT

I TABLE

ICTRD

(1) RBB:

If RBB is input as other than one,
QUAL will find the quality information for RBB ($R\lambda/D$)

(2) IRYTFF.GT.0 writes far field
to unit IRYTFF

(3) KPLOT.GT.0 plots the far field by
calling IPLOT (KPLOT)

(4) ITABLE = 0 finds quality table and
plots information
= 1 does not do the above

(5) ICTRD is used for ITABLE = 0,
= 0 chooses the optimal focal length
based on the highest 1.0 $R\lambda/D$ quality
about IMAX, then constructs the quality
table based on the better of the two
beam qualities at that focal length
(Default and same as previous).
= 1 calculates quality table about centroid for optimum.
= 2 calculates quality table about IMAX
for optimum.
= 3 finds the optimum value about either
centroid or IMAX chosen for the highest
1.0 $R\lambda/D$ quality.

IDENT FQUAL

•IDENT FQUAL

QUAL

```

•INSERT JITTER.124
C *** UMAT IS THE FAR FIELD CENTERLINE INTENSITY DUE TO A PLANE WAVE
C   APERTURED TO A DIAMETER DM WITH A CONVERGING LENS OF FOCAL LENGTH
C   F APPLIED AT THE NEAR FIELD. THE TOTAL POWER IN THE APERTURED
C   PLANE WAVE IS THE SAME AS THAT OF THE CURRENT CU FIELD.
•INSERT CYCLE9.17
COMMON /STRQWL/ IPHNT

```

GDL

```

•INSERT GDL.120
X .IPHNT
•INSERT GDL.134
C
C   IPHNT IS A FLAG FOR PRINTING NNEG AND WNOW FROM STEP
C   = 0 DON'T PRINT
C   = 1 PRINT (DEFAULT)
•INSERT GDL.541
IPHNT = 1
•INSERT GDL.540
ARC(R,ISTEP,3) = IPHNT
•INSERT GDL.593
IIPHNT = ARC(R,ISTEP,3) * .001
•INSERT GDL.14
COMMON /STRQWL/ IIPHNT

```

QUAL

```

•INSERT JITTER.126
IF(KPLOT.GT.0) WRITE(6,3000) FETITL
3000 FORMAT(1H1,20A4//)
IF(KPLOT.NE.0) CALL IPLOT(KPLOT)
IPHNT = 1
IF(IWYIFF.NE.0) WRITE(IWYIFF) (CU(IX),IX=1,NOR),X,DMX,DY,NIT,SAVE
IF(IWYIFF.NE.0) REWIND IWYIFF
IF(IWYIFF.NE.0) WRITE(6,400) IWYIFF
400 FORMAT(10X,'FAR FIELD HAS BEEN WRITTEN TO UNIT',I4)

```

STEP

```

•INSERT STEP.4
COMMON /STRQWL/ IPHNT

```

CAVITY

```

•INSERT CAVITY.9
COMMON /STRQWL/ IPHNT
•INSERT CAVITY.102
IPHNT = 1

```

QUAL

```

•INSERT JITTER.104
DATA FETITL /14*4H .4HFAR .4HFIEL.4HD PL.4HOTS .2*4H /
DATA NFITL /14*4H .4HNPFI.4HMIZF.4HD FI.4HFLD .2*4H /
DATA SAVE /10*0./
IPHNT = 0
DM = 1.0
DMX = 5.0
•INSERT QUAL.14
C *** ISAVE = 0 : READ IN FAR FIELD FROM UNIT 9.
•INSERT QUAL.14
C *** ISAVE = 1 : SAVE NEAR FIELD ON UNIT 7.

•INSERT QUAL.21
C *** ISAVE = -1 : READ NEAR FIELD FROM UNIT 9.
•INSERT CYCLE9.26
C *** WRITE CU FIELD WITH LENS APPLIED (FOCAL LENGTH F) TO UNIT 1.
•DELETE QUAL.112

```

```

CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RH1,PRH)
IF(ISTEP.F0.6) CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RH5,PRH5)
IF(ISTEP.F0.5.AND.RH.NF.1.)
X CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RH,PRRRH)
*DELETE QUAL.121,QUAL.127
WRITE(6,132) RH1,PRH,XCINT,YCINT,RH1,PRH,UMXK,XPEAK,YPEAK,
X PWSAVK,DR,STREHL
IF(RH.NF.1.)
X *IF(6,132) RH,PRRRH,XCINT,YCINT,RH,PRRRH,UMXK,XPEAK,YPEAK,
X PWSAVK,DR,STREHL
132 FORMAT(//15H DCALC FLUX IN .F5.2.6H RL/D=.G12.4.* ABOUT CENTROID*,
X12X.11HCOORDINATES.2G12.4/15H DCALC FLUX IN .F5.2.6H RL/D=.G12.4.
X14H ABOUT IMAX OF.G12.4.12H COORDINATES.2G12.4/13H TOTAL DCALC .
X5HFLUX=.G12.4.5X.22H REFERENCE DIAMETER=.F6.2./
X 19H STREHL INTENSITY =.G11.4/)
HQ5INT = PRH5K/PWSAVK*100.
HQ5CNT = PRH5/PWSAVK*100.
WRITE(6,6010) HQ5INT,HQ5CNT
6010 FORMAT(//10X.*NOTE: CENTROID AND IMAX COORDINATES ARE IN*.
X * CENTIMETERS*//.* HQ ABOUT IMAX FOR 5HL/D=.G12.4.
X 10X.* HQ ABOUT THE CENTROID FORM 5HL/D=.G12.4/)
*DELETE QUAL.118
CALL POWWOW(NPTS,DX,X,US,XCINT,YCINT,RH1,PRH)
ZLDSQ = ZLD*ZLD)
*INSERT QUAL.120
IF(ISTEP.NF.6) GO TO 2000
C *** FIND POWER IN 5HL/D.
CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RH5,PRH5)
CALL POWWOW(NPTS,DX,X,US,XCINT,YCINT,RH5,PRH5)
PRH5 = PRH5/ZLDSQ
PRH5K = PRH5/1000.
PRH5 = PRH5/ZLDSQ
PRH5K = PRH5/1000.
IF(RH.F0.1.) GO TO 2000
C *** FIND POWER IN 4HL/D)
CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RH,PRRRH)
CALL POWWOW(NPTS,DX,X,US,XCINT,YCINT,RH,PRRRH)
PRRRH = PRRRH/ZLDSQ
PRRRHK = PRRRH/1000.
PRRRH = PRRRH/ZLDSQ
PRRRHK = PRRRH/1000.
C *** RETURN TO CENTIMETERS FOR OUTPUT.
2000 XCINT = XCINT*ZLD
YCINT = YCINT*ZLD
XPEAK = XPEAK*ZLD
YPEAK = YPEAK*ZLD
RH(ISTEP) = PRH
HQINT = PRH/PWSAVK*100.
HOCNT = PRH/PWSAVK*100.
IF(UNOP.NF.0.0) GO TO 340

*DELETE CYCLE9.44,CYCLE9.44
IF(ISTEP.F0.1) WRITE(6,5910)
5910 FORMAT(//33X.30H FLUX(HQ) IN 1*RL/D ABOUT .//
X 12H TRIAL FOCAL.4X.5H TOTAL.4X.30H IMAX CENT .
X 4X.4H STREHL .//
X 12H LENGTHS .4X.5H FLUX.4X.30H (XPEAK,YPEAK) (XCINT,YCINT) .
X 4X.4H INTENSITY/1X.79(14=)
WRITE(6,5920) (STEP,F,PWSAVK,PRH,HQINT,PRH,HOCNT,STREHL,
X XPEAK,YPEAK,XCINT,YCINT)
5920 FORMAT(//3H F.11.17=.G12.4.2X.F7.2.7X.1H .F7.2.1H(.F4.1.1H).2X.
X F7.2.1H(.F4.1.2H) .4X.F0.4/33X.1H(.F4.3.1H.F4.3.1H).

```

```

      X IX, IH, (FA, 3, IH, FA, 3, IH)
*DELETE CYCLE9, 17
      X FHM(5), P(6), PR(6), XSAVE(12H), FETITL(20), NFTITL(20), SAVE(10)
*DELETE CYCLE9, 61, CYCLE9, 65
      PORTH = -100.
C *** FIND LOCATION OF MAXIMUM QUALITY ABOUT IMAX.
      DO 370 I=1, 5
      IF(P(I).LE.PORT) GO TO 370
      PORT = P(I)
      ISV = I
370 CONTINUE
C *** FIND LOCATION OF MAXIMUM QUALITY ABOUT THE CENTROID.
      DO 375 I=1, 5
      IF(PR(I).LE.PORTH) GO TO 375
      PORTH = PR(I)
      ISVH = I
375 CONTINUE
C *** DETERMINE FOCAL LENGTH FOR OPTIMAL CALCULATION
      IF(ICTRD.EQ.0.OR.(ICTRD.EQ.2) IOPT=ISV
      IF(ICTRD.EQ.1) IOPT=ISVH
      IF(ICTRD.NE.4) GO TO 380
      IOPT = ISV
      IF(PORTH.GT.PORT) IOPT = ISVH
380 FOPT=FHM(IOPT)
*DELETE QUAL, 131
      IF(ICTRD.EQ.1.OR.(ICTRD.EQ.1.OR.(ICTRD.EQ.0).AND.PRR.GT.PRR))
      X GO TO 53
*DELETE QUAL, 136
53 IF (ITABLE.EQ.1) GO TO 345
      WRITE(6,55) XCINT, YCINT, F
55 FORMAT(1/2X, ' THE FOLLOWING QUALITY TABLE IS FOUND ABOUT',
      X ' COORDINATES (*.612,4, IH, *.612,4,*) FOR F = *.612,4)
      XCINT = XCINT/ZLO
      YCINT = YCINT/ZLO
      CALL PLOT(NPTS, DX, X, UMAX, 4, US, IPLT.
*DELETE JITTER, 103
      SUBROUTINE QUAL (IPHASE, ISAVE, IPLT, TITLE, RH, ANS, OR, RF, SIGANG, PROP
      X KPLT, IWRITE, ITABLE, ICTRD, IRYTFF, NIT)
MAIN
*DELETE JITTER, 2
      NAMELIST/PLT/TITLE, IPLT, OR, ISAV, IPHASE, PRR, RF, SIGANG, PROP,
      X KPLT, IWRITE, ITABLE, ICTRD, IRYTFF
*INSLUT MAIN, 80
C      PROP = 0. PERFORMS FOCAL LENGTH OPTIMIZATION

C      PROP.GT.0. CALCULATES QUALITY FOR THE NOMINAL FOCAL LENGTH ONLY.
C      PROP.LT.0. CALCULATES QUALITY FOR THE CHOSEN FOCAL LENGTH
C              (F = -PROP) ONLY.
C      IWRITE.GT.0 SETS IX = IWRITE
C      IWRITE.LT.0 SETS IX = -IWRITE AND RETURNS TO SQQ IMMEDIATELY.
C      IRYTFF.GT.0 WRITES THE FAR FIELD TO UNIT IRYTFF
C      KPLT.GT.0 PLOTS THE FAR FIELD BY CALLING IPLOT(KPLT)
C      ITABLE = 0 FINDS QUALITY TABLE AND PLOTS INFORMATION
C              = 1 DOES NOT DO THE ABOVE
C      ICTRD IS USED FOR ITABLE = 0.
C              = 0 CHOOSES THE OPTIMAL FOCAL LENGTH BASED ON THE HIGHEST
C              IMAX QUALITY ABOUT IMAX. THEN CONSTRUCTS THE QUALITY
C              TABLE BASED ON THE BETTER OF THE TWO BEAM QUALITIES.
C              AT THAT FOCAL LENGTH. (DEFAULT AND SAME AS PREVIOUS)
C              = 1 CALCULATES QUALITY TABLE ABOUT CENTROID FOR OPTIMUM.
C              = 2 CALCULATES QUALITY TABLE ABOUT IMAX FOR OPTIMUM.
C              = 3 FIND THE OPTIMUM VALUE ABOUT EITHER CENTROID OR IMAX
C              CHOSEN FOR THE HIGHEST IMAX QUALITY.

```

```

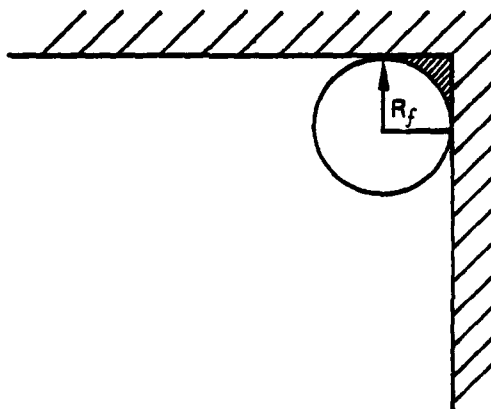
*DELFT JITFW.3
210 CALL DUAL (IPHASE,ISAV,IQLT,TITLE,PHH,AS,DH,WF,SIGANG,PROP,
X KPLOT,IWRITE,ITABLE,ICTRD,IYTFE,N(I)
PPOP = 0.0
KPLOT = 0
PH = 1.
IWRITE = 0
IYTFE = 0
ITABLE = 0
ICTRD = 0
IFLGAP = 1
*INSERT MAIN.22
DATA SIGANG /0.0/
DATA PPOP /0.0/
DATA KPLOT,IWRITE,IYTFE /0.0,0,0/
DATA ITABLE,ICTRD /0.0/
PLTOT
*INSERT PLOT.53
WRITE(40,2000) TITLE
2000 FORMAT(1X,20A4.//,4X,4H/D*,5X,*FRAC1ION*/)
*INSERT APP26.13
DO 2025 I=1,30
2025 WRITE(40,2024) QND(I),PWA(I)
2024 FORMAT(3X,F4.1,5X,F4.5)

```

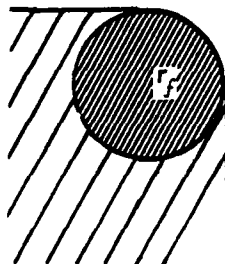
11. *ID FILAPR

Ident FILAPR increases the generality of the aperture routine APRTR by adding filleted apertures.

The outer fillet works by putting a circle in each of the four corners of radius R_f (input as RFOUT or RFMOUT).



The lightly shaded region is removed by a regular rectangular aperture, while the heavily-shaded portion indicates the region removed by the fillet. A central obscuration can be applied in a similar fashion. The result is a rectangular aperture with rounded corners:



The input was a list of names for RFIN for namelist APTUR and RFIN for namelist MIROR.

Subroutine APRTR has also been modified such that it now prints out maximum intensity on both the central obscuration, as well as on outer aperture.

The bare resonator normalization aperture has been generalized to include the fillet as well as being any particular mirror number IM.

IDENT FILAPR

*IDENT FILAPR

TO FILAPR INCORPORATES THE UPDATES FROM PWA/GPD IN WEST PALM BEACH, FLORIDA TO ADD TO THE ABILITY OF THE APERTURE ROUTINE TO APPLY A FILLETED APERTURE.

APRTR

```
*DELETE APRFIX.21
  AMXIN = 0.
  AMXOUT = 0.
  XMXIN = 0.
  XMXOUT = 0.
  YMXIN = 0.
  YMXOUT = 0.
*DELETE APRFIX.51,APRFIX.54
  IF (IIN.FU.1) GO TO 50
C ***** IIN = 0 FOR OUTER APERTURE
  AMXOUT = AMAXI(AFIX,AMXOUT)
  IF (AFIX.NF,AMXOUT) GO TO 60
  XMXOUT = X
  YMXOUT = Y
  GO TO 60
```

```

C ***** IIN = 1 FOR INNER APERTURE
70 AMXIN = AMAX1(AINT,AMXIN)
   IF(AINT.NE.AMXIN) GO TO 60
   XMATN = X
   YMATN = Y
80 CONTINUE
*DELETE APPFIX,88,APWFIX,91
   IF(IIN.EQ.1) GO TO 70
C ***** IIN = 0 FOR OUTER APERTURE
   AMXOUT = AMAX1(AINT,AMXOUT)
   IF(AINT.NE.AMXOUT) GO TO 40
   XMOUT = X
   YMAOUT = Y
   GO TO 40
C ***** IIN = 1 FOR INNER APERTURE
70 AMXIN = AMAX1(AINT,AMXIN)
   IF(AINT.NE.AMXIN) GO TO 40
   XMATN = X
   YMATN = Y
80 CONTINUE
*DELETE APPFIX,102,APWFIX,105
   AMXIN = AMXIN*FXF/1000.
   AMXOUT = AMXOUT*FXF/1000.
   IF(WDISK.NE.0..00,YDISK.NE.0.) WRITE(6,310) AMXIN,AMXIN,YMATN
310 FORMAT(* THE MAX INTENSITY ON THE INNER APERTURE PLATE IS*,
   C * IMAX = *.G13.5/* AND IS LOCATED AT X = *.F13.5.* Y = *.F13.5)
   IF(RAPPTH.NE.0..00,YAPPTH.NE.0.) WRITE(6,320) AMXOUT,XMOUT,YMAOUT
320 FORMAT(* THE MAX INTENSITY ON THE OUTER APERTURE PLATE IS*,
   C * IMAX = *.G13.5/* AND IS LOCATED AT X = *.F13.5.* Y = *.F13.5)
*DELETE APPFIX,1
   SUBROUTINE APPTR(RAPTHI,WDISK,XPOS,YPOS,YAPTH,YDISK,
   X XAPTH,XDISK)
*DELETE APPFIX,2,APWFIX,4

C   MODIFIED 3/4/77 BY P. FILEGER FOR RECTANGULAR APERTURE OF
C   WIDTH=2*XAPTH AND HEIGHT=2*YAPTH AND A CENTRAL
C   ONSCURATION RATIO OF WIDTH=2*WDISK AND HEIGHT=2*YDISK.
C   WHEN RECTANGULAR APERTURES (OR SQUARE) ARE USED, RAPTH AND
C   WDISK BECOME PAIR OF CURVATURE FOR FILLETING THE APERTURE
C   AND CENTRAL ONSCURATION CORNERS RESPECTIVELY.
*INSERT APPFIX,15
   RAPTH = WDHTR
   WDISK = WDISK1
*DELETE SQAPH,3,SQAPH,6
   HO = 2.*YAPTH
   HI = 2.*YDISK
   XO = 2.*XAPTH
   XI = 2.*XDISK
*DELETE SQAPH,8,SQAPH,13
1000 FORMAT(12H CIRCULAR APERTURE APPLIED //
   A 19H OUTSIDE RADIUS = .G12.4/
   A 19H INSIDE RADIUS = .G12.4)
1001 FORMAT(14H RECTANGULAR APERTURE APPLIED //
   A 24H OUTSIDE DIMENSIONS ARE .G12.4.9H HIGH BY .G12.4.5H WIDE/
   A 24H INSIDE DIMENSIONS ARE .G12.4.9H HIGH BY .G12.4.5H WIDE)
   IF(YAPTH.NE.0.) WRITE(6,1004) RAPTH
1004 FORMAT(24H FILLET RADIUS = .G12.4)
   IF(WDISK.NE.0.) WRITE(6,1005) WDISK
1005 FORMAT(24H ONSCURATION RADIUS = .G12.4)
   WRITE(6,1004) XPOS,YPOS
1003 FORMAT(10H XPOS = .G12.4./10H YPOS = .G12.4//)
   IF(YAPTH.EQ.0.0) GO TO 260

```

```

*INSERT SOAPH.1
  WHITE(A.1003) (XPOS,YPOS)
*DELETE APWFIX.32
  IF (IIN.EQ.0.AND.R.GE.WAPHTR) INTCK=1
  IF (IIN.EQ.1.AND.R.LE.WDISK) INTCK=1
*DELETE APWFIX.63,APWFIX.64
  A = XAPHTR
  H = YAPHTR
  AS = A - WAPHTR
  HS = H - WAPHTR
  HAD = WAPHTR
*DELETE APWFIX.70
  IF ((ABS(X).GE.XAPHTR.OR.ABS(Y).GE.YAPHTR).AND.IIN.EQ.0) INTCK = 1
  IF ((ABS(X).LE.XDISK.AND.ABS(Y).LE.YDISK).AND.IIN.EQ.1) INTCK = 1
*INSERT APWFIX.77
  IF (XMIN.GE.AS.AND.YMIN.GE.HS) GO TO 400
*DELETE APWFIX.79
  IF (XMAX.LE.A.OH.YMAX.LE.R) GO TO 200
*DELETE APWFIX.90
  IF (XMAX.LE.A.OH.YMAX.LE.H) GO TO 200
*DELETE APWFIX.94,APWFIX.98
  260 IF (YDISK.EQ.0.OR.IIN.EQ.1) GO TO 300
  IIN = 1
  A = XDISK
  H = YDISK
  AS = A - WDISK
  HS = H - WDISK
  HAD = WDISK
  GO TO 140
100 CONTINUE
  XF = ABS(X) - AS
  YF = ABS(Y) - HS
  XF = SIGN(XF,X)
  YF = SIGN(YF,Y)
  V = SQRT(XF**2 + YF**2)
  IF (R.GE.WAPHTR.AND.IIN.EQ.0) INTCK=1
  IF (IIN.EQ.1.AND.V.LE.WDISK) INTCK=1
  WDP = W(XF,YF,1,1)
  WMM = W(XF,YF,-1,-1)
  WMP = W(XF,YF,-1,1)
  WMI = W(XF,YF,1,-1)
  WFW = 1.
  WMAX = AMAX(WDP,WMM,WMP,WMI)
  IF (WMAX.LE.WAD) GO TO 200
  WFW = 0.
  WMIN = AMIN(WDP,WMM,WMP,WMI)
  IF (WMIN.GE.WAD) GO TO 200
  WFW = (WAD-WMIN)/(WMAX-WMIN)
  GO TO 200

```

GOL

```

*INSERT ROT.4
  Y = YF4IN*WFOUT
*DELETE SOAPH.22
  NAMELIST ZAPHTR, ZOUT, ZIN, XPOS, YPOS, YOUT, YIN, XFIN, XFOUT
*INSERT ROT.3
  DATA DEIN, DEOUT, DEHT, WFOUT /400./
*INSERT GOL.114
  C WEMIN = RADIUS OF CENTRAL ONSCURATION CORNER.
  C WEMOUT = RADIUS OF FILLET.
*INSERT GOL.145
  C WFIN = RADIUS OF CENTRAL ONSCURATION.
  C WFOUT = RADIUS OF FILLET.

```

```

*INSET NAME.33
  RFIN = 0.
  RFOUT = 0.
  RTST = 0.
*INSET NAME.34
  ARC(12,IMR,4) = 0.
  ARC(13,IMR,4) = 0.
  IF (ROUT.NE.0..OR..YI.NE.0.) RTST = 1.
  IF (RTST.EQ.0) GO TO 22
  ARC(4,IMR,2) = RFOUT
  ARC(5,IMR,2) = RFIN
  ARC(12,IMR,4) = 0.5*RT/2.
  ARC(13,IMR,4) = RTAIN/2.
*DELETE NAME.40,NAME.61
  IN = ARC(8,1,1) + .01
  CALL APRTW (ARC(4,IM,2), ARC(5,IM,2), ARC(6,IM,2), ARC(7,IM,2),
    & ARC(10,IM,4), ARC(11,IM,4), ARC(12,IM,4), ARC(13,IM,4))
C *** THE ABOVE ASSUMES THAT CVM IS MIRROR NUMBER IN.
*DELETE NAME.44,NAME.62
  21 CALL MIRROR (ARC(1,IMR,2), ARC(2,IMR,2), ARC(3,IMR,2),
    & ARC(4,IMR,2), ARC(5,IMR,2), ARC(6,IMR,2), ARC(7,IMR,2),
    & ARC(8,IMR,2), ARC(9,IMR,2), ARC(10,IMR,2), ARC(11,IMR,2),
    & ARC(12,IMR,2), ARC(13,IMR,2), ARC(14,IMR,2),
    & ARC(10,IMR,4), ARC(11,IMR,4), ARC(12,IMR,4), ARC(13,IMR,4))
MIRROR
*DELETE NAME.38
  CALL APRTW (RROUT,RTAIN,XPOS,YPOS,RROUT,RYIN,RROUT,RXIN)
*DELETE MIRROR.2,NAME.10
  SUBROUTINE MIRROR (ANX,ANY,PAUC,RROUT,RTAIN,XPOS,YPOS,RFL,DELT,
    & DIST,RANGLS,PHI1ST,PHI1RT,DESI1,
    & RROUT,RYIN,RROUT,RXIN)
C THE FIRST 2 LINES ARE ARC(N,IMR,2) N=1,14 AND
C AND THE LAST LINE IS ARC(N,IMR,4) N=10,13
*DELETE MIRROR.13
  IF (RROUT.EQ.0..AND..RTAIN.EQ.0..AND..RROUT.EQ.0..AND..RYIN.EQ.0.)
    & GO TO 70
END
*DELETE NAME.446
  ROUTY = ARC(10,IMR,4)
  ROUTX = ARC(4,IMR,2)
  IF (ROUTY.NE.0) ROUTY = ARC(12,IMR,4)
  ROUTX = ROUTY
  IF (ROUTY.NE.0) ROUTX = AMT(1)(ROUTX,ROUTY)
*INSET NAME.614
  RFIN = 0.
  RFOUT = 0.
  RTST = 0.
*INSET NAME.14
  ARC(7,AP,4) = 0.
  ARC(8,AP,4) = 0.
  IF (ROUT.NE.0..OR..YI.NE.0.) RTST = 1.
  IF (RTST.EQ.0) GO TO 41
  ARC(1,AP,4) = RFOUT
  ARC(2,AP,4) = RFIN
  ARC(7,AP,4) = 0.5*RT/2.
  ARC(8,AP,4) = RT/2.
*DELETE NAME.34,NAME.624
  IF (ROUT.NE.0..OR..YI.NE.0..OR..GE.0.0)
    & CALL APRTW (ARC(1,AP,4), ARC(2,AP,4), ARC(3,AP,4), ARC(4,AP,4),
    & ARC(5,AP,4), ARC(6,AP,4), ARC(7,AP,4), ARC(8,AP,4))
    & SAVE = ROUTX

```

```

      IF(YOUT.NE.0.) WAPTR = AMPL(DDUT,YOUT)/2.
      IF(YOUT.EQ.0.) WAPTR = DDUT/2.
      IF(WAPTR.LE.0.0) WAPTR = WSAVE
*INFLETH GUL.622
      41 PPW=0.
      DO 13 IZZ = 1,NOR
      13 PPW = PPW + C1(IZZ)*CONJG(CU(IZZ))
      SPPW = PPW*(X(2)-X(1))*2*(NPTS/NPY)
      IF(NWFG.EQ.1.02,NWFG.EQ.2) SPPW=SPPW/WNUW**2
      DDUT = AHC(1,IAP,4)*2.
      DIN = AHC(2,IAP,4)*2.
      YOUT = AHC(5,IAP,4)*2.
      YIN = AHC(6,IAP,4)*2.
      IF(YOUT.NE.0.,OR.YIN.NE.0.) DDUT= AHC(7,IAP,4)*2.
      IF(YOUT.NE.0.,OR.YIN.NE.0.) DIN= AHC(8,IAP,4)*2.
      IF(DDUT.LT.0.0.AND.)DIN.LT.0.0)

```

12. *ID NUDISK

Ident NUDISK modifies the two I/O IFLOWS in GDL, IFLOW = 10 and IFLOW = 16.

- a. IFLOW = 10 Two new options have been added to this IFLOW. Multiple fields can now be written to the same file by not rewinding the file between writes (RWIND = .F.). A file can also be written that can read at a terminal (READS = .T.). For this can the file is written in the following order:

TITLE, NPTS, NPY

(X, [I], I = 1, NPTS)

DO 141, J = 1, NPTS

141 WRITE (IWRITE) (CU[I + (J-1) *NPTS], I =
1, NPTS)

Symmetric fields are unfolded before being written to tape for READS = .T.

- b. IFLOW = 16 This IFLOW has been updated so that formatted data can be read in as well as written out. The format has been modified to include more digits.

IDENT 0017SKI

*IDENT 0017SAT

GOL

```

*DELETE GOL.75
C      = 15  ON PUNCHED ON CARDS. READS PUNCH.
*INSERT GOL.84
      DATA KWRITE,KREAD,KHEAD /0.0.0.F./
*INSERT GOL.184
      NAMELIST /PUNCH/ KREAD,KWRITE
C      THIS IS A FORMATTED VERSION OF DISK1.
C      KREAD IS UNIT TO BE READ FROM - IF ZERO, DON'T READ.
C      KWRITE IS UNIT TO BE WRITTEN TO - IF ZERO, DON'T WRITE.
C
*DELETE GOL.392,GOL.394
      IAO KREAD = 0
      KWRITE = 0
      READ(IN,PUNCH)
      IF(KREAD.EQ.0.AND.KWRITE.EQ.0) GO TO 444
      IF(KREAD.EQ.0) GO TO 169
      READ(KREAD,164) TITLE
      165 FORMAT(20A4)
      WRITE(0,166) KREAD,TITLE
      166 FORMAT(2X,0FORMATTED FIELD READ IN FROM UNIT :*,I3,*,*/1X,20A4)
      DO 167 J=1,NPY
      IREF=(J-1)*NPTS
      DO 167 I=1,NPTS*2
      READ(KREAD,168) X(I),X(J),DUM1,DUMF1,X(I+1),X(J),DUM2,DUMF2
      II = 2*(I+IREF)
      CUM(II-1) = DUM1
      CUM(II) = DUMF1
      CUM(II+1) = DUM2
      CUM(II+2) = DUMF2
      167 CONTINUE
      164 FORMAT(2FH,3.2F12.6,2FH,3.2F12.6)
      REWIND KREAD
      GO TO 444
      169 WRITE(0,169) KWRITE
      163 FORMAT(2X,0FORMATTED FIELD WRITTEN TO UNIT :*,I3,*,*)
      WRITE(KWRITE,164) (GNOT(ICNTL,I),I=1,20)
*DELETE GOL.401,GOL.404
      161 WRITE(KWRITE,168) X(I),X(J),DUM1,DUMF1,X(I+1),X(J),DUM2,DUMF2
      REWIND KWRITE
*DELETE GOL.174
      NAMELIST /DISK1/ IREAD,IWRITE,IORD,IAND,READ3,WRITE3
C      READ3 = .T. MEANS READ OR WRITE TO TAPE IN THREE STEPS.
C      WRITE3 = .T. MEANS REWIND WRITTEN(HEAD) TAPE.
*INSERT GOL.455
      READ3(NOS) = READ3
      WRITE3(NOS) = WRITE3
      IF(READ3(NOS)) GO TO 104
*INSERT GOL.460
      IF(WRITE3(NOS)) GO TO 104
*INSERT GOL.450
      READ3 = .F.
      WRITE3 = .T.
*DELETE GOL.462
      I2 = IREAD
      IF(IREAD.LT.0) I2 = -IREAD
      IF(IREAD.GT.0) READ (IR) (CU(I2),I2=1,NOR),X,DW,DWY,NITER
      IF(IREAD.LT.0) READ (IR) (CU(I2),I2=1,NOR),X,DW,DWY,NITER,SAVE

```

```

*DELETE GOL.472
  IR = IREAD
  IF (IREAD.LT.0) IR = -IREAD
  IF (IREAD.GT.0) READ (IR) (CU(IZ).IZ=1,NOM).A.DHX.DRY.NITER
  IF (IREAD.LT.0) READ (IR) (CU(IZ).IZ=1,NOM).A.DHX.DRY.NITER.SAVE
*DELETE GOL.466
  IF (RWIND) REWIND IREAD
  IF (.NOT.RWIND) WRITE(6,104)
*DELETE GOL.470
  IF (RWIND) REWIND IWRITE
  IF (.NOT.RWIND) WRITE(6,104)
*DELETE GOL.474
  IF (RWIND) REWIND IREAD
  IF (.NOT.RWIND) WRITE(6,104)
104 FORMAT(10X,'THE FILE HAS NOT BEEN REWOUND')
*INSERT GOL.475
108 IF (IREAD.EQ.0.AND.IWRITE.EQ.0) GO TO 999
  IF (IREAD.EQ.0) GO TO 110
  READ(IREAD) TITLE.NPXIN.NPYIN
  DO 120 I=1,20
120 DTITLE(ND5,I) = TITLE(I)
  WRITE(6,121) IREAD.NPXIN.NPYIN.TITLE
121 FORMAT(2X,'PARTITIONED TAPE BEING READ FROM UNIT :*.I3.*.*
  X * WITH NPXIN =*.I5.10X.*AND NPYIN =*.I5.
  X /1X.20A4/)
  READ(IREAD) (X(I).J=1.NPXIN)
  DO 123 J=1.NPXIN
  JMI = (J-1)*NPXIN
123 READ(IREAD) (CU(I+JMI).I=1.NPXIN)
  NPTS = NPXIN
  NPY = NPYIN
  IF (RWIND) REWIND IREAD
  IF (.NOT.RWIND) WRITE(6,104)
  GO TO 999
110 READ(IN,124) TITLE
  DO 125 L=1,20
125 DTITLE(ND5,L) = TITLE(L)
  WRITE(6,126) IWRITE.TITLE
126 FORMAT(2X,'PARTITIONED TAPE BEING WRITTEN TO UNIT :*.I3.*.*
  X /1X.20A4/)
  WRITE(IWRITE) TITLE.NPTS.NPY
  WRITE(IWRITE) (X(I).J=1.NPTS)
  IF (NPTS.EQ.NPY) GO TO 140
C *** UNFOLD CU
  DO 130 J=1.NPY
  JMI = (J-1)*NPTS
  JIM1 = (NPTS-J)*NPTS
  DO 130 I=1.NPTS
  IJ = I+JMI
  IJ1 = I+JIM1
130 CU(IJ) = CU(IJ1)

140 DO 141 J=1.NPTS
  JMI = (J-1)*NPTS
141 WRITE(IWRITE) (CU(I+JMI).I=1.NPTS)
  IF (RWIND) REWIND IWRITE
  IF (.NOT.RWIND) WRITE(6,104)
  GO TO 999

```

13. *ID CY4KIN

The capabilities of the numerical SOQ kinetics package have been expanded to include oxygen, hydrogen, and R-branch transitions (9.4 μ band).

- a. Oxygen has been upgraded from a structureless molecule to one that has structure. Therefore, there are now kinetics rate equations for the interaction of oxygen with the rest of the molecules from the combustion process.
- b. Hydrogen has been included as a structureless collision partner.
- c. Previously the code has used P-Branch transitions (10.4 μ). Using GFACT less than 1 now activates the 9.4 μ R-Branch transition.

In addition to the above major changes, two small additions have been made:

- (1) Input the Gladstone-Dale constant GDC in name-list CAVTY2.
- (2) Account for the gain length by the factor ZFACT, also in CAVTY2.

IDENT CY4KIN

*IDENT CY4KIN

TO CY4KIN INCORPORATES THE CYCLE IV KINETICS PACKAGE FROM
SWAZEM IN WEST PALM BEACH, FLORIDA. OXYGEN AND HYDROGEN
KINETICS ARE ADDED AS WELL AS THE ABILITY TO STIMULATE
R-BRANCH TRANSITIONS OF THE 9.4 MICRON BAND.

HLIMIT

*DELETE HLIMIT.11

4 TITLE(20),AVG(5),VOP(5),FM2(5),NSYM

CAVITY

*DELETE CAVITY.16

4 AVG(5),VOP(5),FM2(5),NSYM

*INSERT LPOP1.10

DATA THE 20.2

*DELETE LPOP1.11

4 AVGAIN,GFACT2,1.12,XXX,GDC,ZFACT

*DELETE CAVITY.75,CAVITY.77

C 11 IS VIBRATIONAL TEMPERATURE OF VUO AT NEP. DEG K

C 12 IS VIBRATIONAL TEMPERATURE OF VUO AT NEP. DEG K

C 13 IS VIBRATIONAL TEMPERATURE OF VUO AT NEP. DEG K

*DELETE CAVITY.82

C BANC1 IS THE J VALUE OF THE LOWER LASER LEVEL FOR THE TRANSITION

```

*INSERT CAVITY.127
C      AM2 IS MOLE FRACTION OF HYDROGEN
*DELETE CAVITY.127
WRITE (6,103) X12,XC02,XH20,XC0,X02,XH2,GDC
*DELETE CAVITY.129
X AMH20 = .G12.5+.4X.5HX00 = .G12.5+.4X.5HX02 = .G12.5+.4X.5HXH2 =
X .G12.5+.4 GLAUSTONE=DALE CONSTANT = .G12.5)
*INSERT CAVITY.136
IV02(NCAVN)=T02
*INSERT CAVITY.138
WRITE (6,1104) T02
1104 FORMAT(1X,A,F12.5)
*INSERT CAVITY.174
IF (X02.NF.0..AND..T02.EQ.0.) STOP "HFFD T02"
FM2(NCAVN) = XH2
DENSITY
*INSERT DENSITY.30
COMMON /GLAD/ STONE,ZFACTW
*INSERT DENSITY.100
GDC = STONE
CAVITY
*INSERT CAVITY.9
COMMON /GLAD/ STONE,ZFACTW
GAINXY
*INSERT GAINXY.15
COMMON /GLAD/ STONE,ZFACTW
*INSERT GAINXY.114
DELTAZ = DELTAZ*ZFACTW
IF (ZFACTW.NF.1.) WRITE (6,1000) ZFACTW,DELTAZ
1000 FORMAT(2SA.5H000000.0 WARNING = GAIN MODIFIED BY*.F7.4.
X * ACTIVE LENGTH. DELTAZ =.G12.5.5H000000)
CAVITY

*INSERT CAVITY.103
GDC = STONE
ZFACT = ZFACTW
*INSERT CAVITY.109
STONE = GDC
ZFACTW = ZFACT
DENSITY
*DELETE DENSITY.31
FILMS
*DELETE FILMS.11
A      AVG(5),IV02(5),FM2(5),NSYM
COMMON /GLAD/ STONE,ZFACTW
*DELETE FILMS.119
ROCL=STONE*WM0*ZC(NCV)/NS(NCV)
GAINXY
*DELETE GAINXY.10
COMMON/START/ TST,PST,VT,F00VT,F0V01,FV001,END1,GAIN1,FGY021
*DELETE GAINXY.12
COMMON /MOLES/ X12,XC02,XH20,XC0,X02,XH2
*DELETE GAINXY.14
COMMON/PATH/ X12,PC1,WC2,PPUMP,PST14,WH,RH,RH10
*DELETE GAINXY.20
S      TITL(20),AVG(5),IV02(5),FM2(5),NSYM
*INSERT GAINXY.26
AM2 = FM2(NCV)
*INSERT GAINXY.38
T02=IV02(NCV)
*INSERT GAINXY.44
GCON=.441E-14*PM

```

```

      EXF2=-1007.
      DEF=140.
      IF (GEACT(NCV).GT.1.) GO TO 10
C   MODIFY CONSTANTS FOR 2-BRANCH TRANSITION
      ROTUP=.554*(WR+1.)*(WR+2.)
      GCUN=.742*(14*(WR+1.))
      EXF2=-1050.
      DEF=1530.
      10 CONTINUE
      *DELETE GAINXY.44
      *DELETE LQW1.14,LQW1.15
      *DELETE LQW1.16,LQW1.17
      *DELETE LQW1.18,LQW1.19
      XW = 24.016*XM2 + 44.011*XC02 + 14.016*XM20 + 32.*X02 + 2.016*XM2
      X02FAC=18.524
      *INSERT LQW1.20
      X = .34*XM2/5*RT(1.924/22.005)
      *DELETE LQW1.21
      *DELETE GAINXY.56
      XL44 = 1.434/(DEF+ROTUP-ROTLO)
      *DELETE GAINXY.60
      GAMMA = (7.*(X+2*XC02+X02+XM2)+4.*XM20)/(5.*(XM2+XC02+X02+XM2)+
      X = 4.*XM20)
      *DELETE GAINXY.68
      CW = 3.5*W0*(XM2+XC02+X02+XM2+4./7.*XM20)
      *INSERT GAINXY.72

      EGY02 = 0.0
      IF (X02.NE.0.) EGY02=X02*1556./(EXP(2239./T02)-1.)
      *INSERT GAINXY.85
      EXFROT=EXF2/T1
      IF (GEACT(NCV).LT.1.) EXFROT=EXF2/T2
      *DELETE GAINXY.87
      X = -.561*EXP(EXFROT-ROTLO/T5))
KINFT
      *DELETE KINFT.4,KINFT.9
      COMMON/STAMP/ TS,PSI,VT,F00V1,F0V01,FV001,FN21,GAINI,EGY02
      COMMON /MOLFS/ XM2,XC02,XM20,XC0,X02,XM2
      *DELETE KINFT.11
      COMMON/HATE/ XM2,RC3,RC2,NOJAP,NSTIM,RA,RQ,R10
      *INSERT KINFT.12
      COMMON/GEACT4/GEACT(2)
      *INSERT KINFT.23
      F02=X02*1556.
      DEY001 = 0.
      DEY002 = 0.
      CHG4F = 0.
      EXF2=-1007.
      IF (GEACT(1).LT.1.) F4=1.245E10/MNU
      IF (GEACT(1).LT.1.) EXF2=-1950.
      *INSERT KINFT.26
      EGY02=EGY02I
      *INSERT KINFT.71
      IF (X02.NE.0.) GO TO 30
      02FEXP=EXP(-2239.044/T5)
      DE02F02=12*02FEXP/(1.-02FEXP)
      F0V01=-EXP(-456.4/T5)
      F0V02=(E0V1-F0V0)/E0V0
      F0N21=-EXP(-3354.304/T5)
      E0N2=(E0N2-F0N2)/F02
      CHG4F=(EGY02-DE02F02)*W4
      F0N2=1.-02FEXP
      E0N2=(DE02F02-F0V02)/EGY02

```

```

DE0242=94*G02/T02*F02/E02*(TH02*E02-THN2*E02)
DE0202=-1354.303/2234.*DE02N2
DE02=N10*E02/T02*(E0V0/E0V0)**2.*(1.+TH02*E02-(1.+TH0V0*E0V0)
      1**2.)
DE0V01=-2.*959.8/2234.*DE02
CHG02=(-CHG0E+DE02N2+DE02)*DT
E0V02=E0V02+C002
10 CONTINUE
*DELETE KINFT.75,KINFT.74
DEGL=DE0V0-1.034*DE02MP-1.044*DE0V0V-F4*F10*DT*DE0V01*DT
F02=F02-DE02*DE02N2*DT
*DELETE KINFT.70
SUMDEV=SUMDEV+(DE0V0+CHG0E*DT)*V*1.487E-14*RMUN
*DELETE KINFT.43
DE0V=(DE0V0/DT+CHG0E)*1.14677E8
*INSERT KINFT.121
EXFROT=EXFR/T1
IF(GFACT(1).LT.1.) EXFROT=EXFR/T2
*DELETE KINFT.123
      X=-.56)*EXP(EXFROT-KOIL0/TS))
MIX
*DELETE MIX.6,MIX.7
COMMON /MOLFS/ XN2,XC02,XH20,XC0,X02,XH2
COMMON /WATE/ W02,W03,W02,PPUMP,WSI,M08,R9,W10
*DELETE MIX.24,MIX.24
C C02(0V0)+H2 = C02(0V0) + H2
TC3H = EXP(12.4*TT02+4.49*TT02-2.13)
C C02(0V0)+H2 = C02 + H2
TC2H = EXP(112.*TT02-57.2*TT02+1.723)
W02 = W02*(X02/TC2H+XC02/TC2C+XH20/TC2W+X02/TC20+XH2/TC2H)*1.E6
W03 = W03*(XN2/TC3H+XC02/TC3C+XH20/TC3W+X02/TC3N+XH2/TC3H)*1.E6
*INSERT MIX.31
W10=X02/(H.E-H-TS/150.*1.F-H) *WS
W02=XN2/.345E-2 *WS
W03=1./EXP(W04.4/TT02) * (XN2/5.4E-9 + XC02/1.F-H + XH20
      X /C.7E-13 + X02/5.4E-9) *WS
REGAIN
*DELETE REGAIN.13
      TITLE(20),AVG(5),TV02(5),FH2(5),NSYM
SIMPGG
*DELETE SIMPGG.11
      TITLE(20),AVG(5),TV02(5),FH2(5),NSYM
COMMON/GFACT4/GFACT(2)
*INSERT SIMPGG.49
IF(GFACT(1).LT.1.) EIA = 0.45
MAIN
*INSERT MAIN.20
DATA GDC /0.225/.7FACT /0.96/
*INSERT MAIN.143
STONE = GDC
ZFACT = ZFACT
*INSERT MAIN.8
COMMON /GLAD/ STONE,ZFACT

```

14. *MIRFIX

The MIRROR subroutine has been modified to calculate the effect of power-induced surface curvature when mirror reflectivities other than

the design value are used. The parameter, δ , is modified to change the center to edge distortion as a function of the mirror reflectivity. The parameter, RFLFAC, is used to scale δ as input through the relation:

$$\delta' = \delta \left(\frac{1 - R}{1 - R_d} \right) \left(\frac{P}{P_d} \right) \quad (B2)$$

Where

R = Mirror reflectivity

P = Incident energy

d = Design value

Further, the MIRROR routine has been updated to include the calculation of its own value of mirror flux-induced distortion factor when mirrors are encountered off axis as noted by PHIAST $\neq$ 0. This update has not been activated, since it would mean input file changes for all users. It is included in the code and will be activated by each user, when so desired.

IDENT MIRROR

```
*IDENT MIRROR
MIRROR
*DELETE MIRROR.35
  PWRDES = 1.00
  RFLDES = .005
  RFLFAC = (1.-RFL)/(1.-RFLDES)
*DELETE MIRROR.70
  DELTA = DELTA+PWRDES*RFLFAC
*INSERT MIRROR.140
  IF(NREF.EQ.1.OR.NREF.EQ.2) DELL=DELL/WNO**2
*DELETE MIRROR.23
  IF(PHIAST.NE.0.) WRITE(6,420) PHIAST,RMSAG,RMTAN
*DELETE CIOASTG.14
  IF(PHIAST.NE.0.) WRITE(6,420) PHIAST,RMSAG,RMTAN
*DELETE CIOASTG.24
  IF(PHIAST.NE.1.) WRITE(6,420) PHIAST,RMSAG,RMTAN
*DELETE CIOASTG.20
  420 FORMAT(/,---ASTIGMATIC PHASE ABERRATION APPLIED WITH---%,/,
    x 20x,---PHIAST = %,F10.3, DEG,%,/,
```

GOL

```
*INSERT GOL.531
C IF(PHIAST.EQ.0.) GO TO 19
C DISTF = DISTF*(COS(PHIAST*3.141593/180))**2
C WRITE(6,18) DISTF
C 18 FORMAT(/5x,WARNING: DISTF HAS BEEN MODIFIED BY THE SQUARE.
C x 4 OF COS(PHIAST). NEW DISTF = %,G12.5/)
C 19 CONTINUE
```

15. *ID PROP

The SOQ code calculation of far field performance is based on the analytical equivalence between the Fraunhofer pattern and the propagation of a distribution with field curvature, f , a distance $Z = f$, using the Raleigh-Sommerfield formulation of the diffraction integral in the Fresnel degree of approximation. The SOQ far field calculation propagates the wave distribution, CU , a distance f , determined in a manner which preserves the correspondence between near field and far field coordinates, while accurately resolving the energy spectrum in far field coordinates.

In certain cases, however, it has become necessary to propagate the distribution CU to an arbitrary focal plane Z , using the SOQ calculational procedure, in order to obtain the effects of beam jitter at a fixed distance Z and to obtain the far field information scaled to same focal length. Since far field calculations are based on the use of "vacuum" propagation, the far field at any plane Z'' is simply the scaled distribution at any other plane Z' . This can be shown by comparing the far field distributions in terms of the Fresnel integrals at two arbitrary focal planes Z' and Z'' , where a field curvature of $f' = Z'$ and $f'' = Z''$ has been applied to obtain the distribution. Comparison of these two distributions for the same transmitting aperture size leads to the following scaling.

$$CU_{Z''} = CU_{Z'} \cdot \frac{f}{Z} \cdot e^{-ik(f-Z)} \frac{-ik}{2f} \frac{1}{x}^2 \left(1 - \frac{Z}{f}\right) \quad (B3)$$

And

$$\tilde{x}'' = \frac{Z}{f} \tilde{x}' \quad (B4)$$

where f is the propagation focal distance obtained in the usual manner from the SOQ code, and Z is the "new" scaled propagation distance.

These changes are incorporated in the SOQ code primarily in subroutine QUAL, as documented by the following Fortran changes.

1000 IF PROP

1010 IF PROP

1020 ADDS THE ABILITY TO PROPAGATE TO A SPECIFIC FOCAL LENGTH FROM SUBROUTINE QUAL.

```

QUAL
*DELETE QUAL.41
IF (PROP.0.0) GO TO 44
C *** PROP.0.0. THATS APPLY A LENS OF FOCAL LENGTH PROP (CONVERGING)
C AND PROPAGATE TO THAT FOCAL LENGTH.
FRATIO = -F/PROP
F = -PROP
ZLO = ZLO/FRATIO
WT = WT*FRATIO*FRATIO
DO 41 I=1,NPTS
XSAVE(I) = XSAVE(I)/FRATIO
41 X(I) = XSAVE(I)
OY = OY/FRATIO
OXSAVE = OXSAVE/FRATIO
WK02 = WK/2.
DO 42 J=1,NPY
JM1 = (J-1)*NPTS
YSJ = A(J)*2
DO 42 I=1,NPTS
II(J) = 2*(1+JM1)
II(J+1) = I*J - 1
OYI = WK02*(X(I)**2 + YSJ)*(FRATIO-1.)/PROP
C ** RECALL THAT PROP IS NEGATIVE
COSP = COS(PHI)
SINP = SIN(PHI)
COWE = COS(WT*JM1)
COWI = COS(WT*I)
COWEIJ(JM1) = (COWE*COSP + COWI*SINP)*FRATIO
42 COWEIJ(J) = (COWE*SINP + COWI*COSP)*FRATIO
45 CONTINUE
C *** CHANGE A TO DIMENSIONLESS FAR FIELD X BY DIVIDING BY ZLO=1RL/D.
*DELETE JITTER.122
C *** APPLY A SMALL PHASE TO COWI SO THAT ITS PHASE IS NOT IDENTICALLY
C ZERO.
COWI(I/J) = 1.E-10/FLC0AT(I/J)
*INSERT JITTER.114
44 CONTINUE
*DELETE JITTER.115
IF (PROP.0.0) GO TO 44
C ***
*DELETE JITTER.47
C *** APPLY A SMALL PHASE TO COWI SO THAT ITS PHASE IS NOT IDENTICALLY
C ZERO.
COWI(I/J) = 1.E-10/FLC0AT(I/J)
QUAL
*INSERT CYCLE9.40
IF (PROP.0.0) ISTEP=
*INSERT CYCLE9.46
IF (PROP.0.0) GO TO 440
*DELETE CYCLE9.51,CYCLE9.52
5004 WRITE(6,5040) F,ZLO
5040 FORMAT(//9X,'CONTINUUM RESULTS AT F =*,G12.4,2X,* WITH *
4 144)*#01AMR020) =*G12.4,144)
440 IF (PROP.0.0) WRITE(6,5041) F,ZLO
5041 FORMAT(//20X,'PROPAGATION RESULTS FOR F =*,G12.4,10X,* WITH *
4 144)*#01AMR020) =*G12.4,144)
16. *ID ECSFIX

```

The updates for ECSFIX are included to correct original errors in dimensioning Level II variables, and to reduce the resident array sizes at load and execution of the code.

IDENT ECSEIX

*IDENT ECSEIX

STEP

*INSERT STEP.7

X .APR

COMMON /STEPCL4/ APR

DENSY

*DELETE DENSY.23.CORR2.5

COMMON /MEL1/ P(20000). X4(21). Y4(21.M1). Z4(21.A1).

X C4(21.A1). M4(21). N4. ROCL. DUMYS(4077H)

CAVITY

*DELETE CORR1.43

*DELETE CLODENS.3

X PDD(2).XCAV(140).COR(3276H).US(17100)

*DELETE CAVITY.22

EQUIVALENCE (CU(1).COW(1)). (CG(1).US(1))

REGAIN

*DELETE SQW77CY1.189

*DELETE CLODENS.13

COMMON /GLAD/ STONE.ZFACT

DIMENSION PDD(17100). P(17100). G(17100)

*DELETE SQW77CY1.190.CLODENS.34

EQUIVALENCE (PDD(1).CG(1)). (G(1).CFIL(1)). (P(1).CU(1))

17. *ID SEGSOQ

The SOQ code, as currently configured, is too large to run on the Cyber 176 under AFWL Small Core Memory (SCM) restrictions (high speed core). The segmented load option of the CDC NOS/BE loader has been used to reduce execution time SCM requirements without loss of generality of the code. A segmentation loader, and the appropriate "tree" structure of the code segmentation is required to take advantage of this feature.

To incorporate this scheme into the SOQ code, the SOQ code required additional GLOBAL commons to save certain values, as described on the following Fortran listing. A segmentation tree was developed and is listed also. Further information on segmentation is available in the CDC/NOS/BE loader reference manual. This approach was selected instead of overlay structure because it is a more powerful tool, even though it is machine specific.

IDENT SEGSOQ

*IDENT SEGSOQ

ID SEGSOQ INCLUDES COMMONS THAT CAN BE SAVED FOR THE
SEGMENTED LOADING OF THE DECK.

DENSY

*INSERT DENSY.22

COMMON /SEGIDEN/ TMA.M5.X14R.MMA.Y1.ZC.YHW.ZHW.

X TMA.M4.X14R.MMA.Y4.Z4.YOW.ZOW

CAVITY

*[INSERT CAVITY.1A

COMMON /SF6CV2/ XLEN,YLEN,ZLEN,XMCAV,YMCAV,NODX,NODY,NOSEG,
X FLAG,MFAST,NGTYPE,NGPLOT,INSE,IPDEN,T1,T2,T3,IN2,IS,PS,V,
X PHMCH,XN2,XD2,XH2,XCO,XOP,ALFA,ACH,VELTY,TTEMP,ANGL,
A AVGAIN,GFACIN,TOP,XH2,P1,NH4,NH5,CAKAY,TOPW

GDL

*[INSERT GDL.1A

COMMON /SFGGDL/ FLOW,GNUTE,IPLOTS,KPLOT,
A NCVND,ILX,NSTF,APLT,ZPNDPI,ZPNDP,
R ANGAX,ANGYY,RAOC,DIADUT,DIATIN,XMPOS,YMPOS,XMIR,RFMIN,RFMOUT,
C DELTA,DISTF,PRUTY,DINY,WANDLS,PHIASI,PHIHUT,DESPW,
D DELZ,ROCKV,WINDOX,WINDOK,TIFG,TITR,TIPS,
E DOUT,DIN,YPOS,YPOS,YOUT,YIN,RFOUT,RFIN,
F DIFAM,DVRLAP,DXKH,DYYH,MAXIT,AVCUSM,CUSMF,NEWNPT,NEWNPY,IMTRSM,
G TITLE,MAPLT,DSM,NEWOF,PHIAP4,H0,H,PERNG,
H ALFA,SCF,T,NH0,ZLEN,NSTEPS,INPT,NHHP,AXIAL,UT,
I IREFD,IWRITE,IOR,IADN,READ3,HWINO,
J TRANS,XMAG,NREAN,AWL,NGH0,JITANG,JITIS,KREAD,KWRITE,
K ALPHAM,COMMIW,ALPHAS,PHOGAS,TAU,TIN,REFMIR,CONGAS,
L ISPD,PLDTH,THEFA,XSPC,YSPC,DITH,CAPW,EXPAND,ROC,DISP,TILT,
M DELZM,DELZTH

MAIN

*DELETE CORR.1,50077CY1.2

PROGRAM S00(OUTPUT=512,TAPE1=512,TAPE2=512,TAPE3=512,TAPE4=512,
X TAPE5=512,TAPE6=OUTPUT,TAPE7=512,TAPE8=512,TAPE9=512,TAPE10=512,
X TAPE11=512,TAPE12=512,TAPE13=512,TAPE14=512,TAPE15=512,
X TAPE16=512,TAPE17=512,TAPE18=512,TAPE19=512,TAPE20=512,
X TAPE21=512,TAPE22=512,TAPE23=512,TAPE24=512,TAPE25=512,
X TAPE26=512,TAPE27=512,TAPE28=512,TAPE29=512,TAPE30=512,
X TAPE31=512,TAPE32=512,TAPE33=512,TAPE34=512,TAPE50=512)

C

C

THIS VERSION OF THE S00 CODE CAN BE RUN USING THE SEGMENTED
LOADER ON THE CYBER 176 COMPUTER. THE CRITICAL JCL IS

C

C

ATTACH,IGD,45077912H,IO=*****.

C

ATTACH,TRF,S00SFGTRF,IO=*****.

C

SFGLOAD, I=TRF.

C

END.

C

THE FILE S00SFGTRF CONTAINS THE FOLLOWING CARDS:

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

GLOBAL FST,MFLT,CAV2,CAVX,CG,RAHES,GLAD
GLOBAL PLTSIG,TNITL,XAY,MWPROP,GFACIN,LENSY
GLOBAL START,PRPT,MOLES,ENERG,RATE,STRWHL
GLOBAL FACTER,DAZ,TIME,VIFW,CG,STPLCM,SVTYM
GLOBAL FCL,C,OH,IO,AOR,WM,CON,WM,PUT,FO,TERM,WM
GLOBAL SKFL,FO,SKSF,FO,JMPS,WM,OPFS,FO
CAVITY GLOBAL SF6CV2-SAVE

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

GLOBAL SFGGDL-SAVE
GLOBAL ZIP-SAVE
GLOBAL S00=(QUAL,SGDL,LISTH0)
GLOBAL S00L=(CAVITY,FIFLOS,MIRROR,REGAIN,RSTEP,SLIVER,
*THLOM,AAICN,ZERN)
INCLUDE CHFR,WM,PTWH,SQ,SKSF,SQ,OPES,SQ
INCLUDE STEP,FOIRT,TILT,HACKSP=EOF,ATAN,SPTAN
INCLUDE ACOSIN=COS=SIN,SINCOS=
GLOBAL INCLUDE INTERP
GLOBAL S00LVP INCLUDE SPIDER
GLOBAL THLOM INCLUDE THERML
END

WHERE THE LEFT HAND COLUMN STARTS IN COLUMN 1.

18. *WINDOW

The aerodynamic window subroutine of GDL is used to model the effect of an aerodynamic window on the propagated field in a Monte-Carlo sense. The aerowindow subroutine simulates a random phase transmission function whose rectangularly distributed random phase information can be selected with arbitrary "strength" or variance. This version of AEROW is designed to simulate the phase field degradation with rectangular probability distribution in phase of 0.25λ .

IDENT WINDOW

*IDENT WINDOW
GDL

*DELETE GDL.481
CALL AEROW

AEROW

*DELETE AEROW.2

SUBROUTINE AEROW

*DELETE AEROW.6.AEROW.7

COMMON/AFELT/CU(16384),CFIL(16512),X(128),WL,NPTS,DMX,DY
COMPLEX CU,CFIL

*INSERT AEROW.3

LEVEL 2,00

*DELETE AEROW.9.AEROW.10

PMX=0.

CNT=0

TWPI=6.2831853

SIGMAM = TWPI * .25

NOM=NPTS*NPTS

*DELETE AEROW.12.AEROW.21

P=RAJF(1)*SIGMAM

CNT=CNT + 1

IF(P.LE.PMX) GO TO 20

PMX = P

20 CONTINUE

1 CU(1) = CU(1)*CFIL(CMPLEX(0.,-P))

*INSERT AEROW.22

WRITE(6,100)PMX,CNT

100 FORMAT(20X,'MAX PHASE SHIFT= ',F15.7,'RADIANS',E15.6,'/')

AFWL-TR-79-141

DISTRIBUTION

AUL/LDE, Maxwell AFB AL
DTIC/DDA, Cameron Sta, Alexandria VA
AFWL, Kirtland AFB NM
 (SUL)
 (HO)
 (ARLO)
AFSC/DLWM, Andrews AFB DC
AFELM, Rand Corp, Santa Monica CA
 (Lib D)
Official Record Copy (Capt J. D. Holt/ARLO)

DAT
ILM