AD-A103 286 UNITED TECHNOLOGIES CORP WEST PALM BEACH FLA F/G 20/5
SYSTEM OPTICAL QUALITY USERS GUIDE. PART 3.(U)
MAR 80 J L FORGHAM, S S TOWNSEND F29601-77-C-0025
UNCLASSIFIED AFWL-TR-79-141-PT-3 NL

1 OF 1
AD A
103286

END
DATE
FILMED
10-81
DTIC

AD-E200 724

AFWL-TR-79-141, Pt. 3

(2) LEVEL III

AFWL-TR-79-141 Pt. 3

AD A103286

SYSTEM OPTICAL QUALITY USERS GUIDE

Part 3 ~~of 3~~

J.L. Forgham G. S. Townsend J.L. Campbell

United Technologies Corporation
West Palm Beach, FL 33402

Mar[redacted]980

DTIC ELECTE AUG 2 4 1981 S D B

Final Report

Approved for public release; distribution unlimited.

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

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.

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 PL,ID = AFLOJRA as a NOS/BE-1 CDC update format file.

UPDATES

1. *ID FIXZRN

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

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

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

2. *ID FIXJTR

This update ensures a correct definition of DF in subroutine JITRBG since when JITRBG 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).

COSROT = COS(θ)

SINROT = SIN(θ)

XROT = XIN x COSROT + YIN x SINROT

YROT = -XIN x SINROT + YIN x 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 ZRNIKE.115
           DEL = DEL*3.14159264
     *DELETE ZRNIKE.125
           C 2(X,22H PHI(N) = PI*F(N)*Z(N)//

*IDENT FIXJTR
   */  JITREG
     *DELETE JITTER.29,JITTER.30
           DF = 1./(FLOAT(NPTS)*DX)

*IDENT ROTZRN
   */  SCL
     *DELETE ZRNINFO.3
           NAMELIST /ZERNS/ RO,F,FFRNG,SIGMAY,NTERMZ,ZTHETA,FLIPX,FLIPY
     *INSERT ZRNIKE.5
     C            ZTHETA = THE CLOCKWISE ANGLE OF ROTATION OF THE DECOMPOSITI
     C                     AXES ONTO THE SOG COORDINATE SYSTEM
     C                     BEFORE CALCULATION OF THE ZERNIKE POLYNOMIALS.
     C                     IT IS INPUT IN DEGREES.
     C            FLIPX = 1. RESULTS IN A FLIP ABOUT THE X AXIS BEFORE
     C                     ROTATION.
     C            FLIPY = 1. RESULTS IN A FLIP ABOUT THE Y AXIS BEFORE
     C                     ROTATION.
     *DELETE ZRNINFO.2
           DIMENSION PZZSV(20,10)
     *INSERT ZRNINFO.7
           ZTHETA = 0.
           FLIPX = 0.
           FLIPY = 0.
     *INSERT ZRNINFO.9
           PZZSV(IZERN,3) = ZTHETA*3.141593/180.
           PZZSV(IZERN,4) = FLIPX
           PZZSV(IZERN,5) = FLIPY
     *DELETE ZRNINFO.10,ZRNINFO.11
       244 CALL ZERN(PZZSV(IZERN,1),PZZSV(IZERN,2),PZZSV(IZERN,3),
          X           PZZSV(IZERN,4),PZZSV(IZERN,5),
          X           PZSAVE(25,IZERN),PZSAVE(1,IZERN))
   */  ZERN
     *DELETE ZRNINFO.12
           SUBROUTINE ZERN(SIGMAY,XNTERMZ,THETA,FLIPX,FLIPY,RO,F)
     *INSERT ZRNIKE.72
           COSROT = COS(THETA)
           SINROT = SIN(THETA)
     *DELETE ZRNIKE.75
     *DELETE ZRNIKE.77
           XIN = X(IX)
           YIN = X(IY)
           IF(FLIPX.GT..5) YIN=-YIN
           IF(FLIPY.GT..5) XIN=-XIN
           XROT =  XIN*COSROT + YIN*SINROT
           YROT = -XIN*SINROT + YIN*COSROT
           IF(FLIPX.LT.-.5) YROT=-YROT
           IF(FLIPY.LT.-.5) XROT=-XROT
           XSQ = XROT**2
           YSQ = YROT**2
     *DELETE ZRNIKE.80
           THET = ATAN2(YROT,XROT)
```

```
*IDENT MCRSUM
      *INSERT SUMMRY.515
      C
      C   **** COPY TAPE(50) TO OUTPUT:
      C
            END FILE 50
      C
            WRITE(6,3035)
            REWIND 50
       7000 READ(50,4005) IC1,C2
       4005 FORMAT(I1,21A4)
            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 OUTPUT:
      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 OUTPUT:
      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
      GO TO 5000
 5015 REWIND 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
Code
Optics
Optical Quality
Optical System
High Power Laser
High Energy Laser

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)

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	
Dist	Avail and/or Special
A	23 CP

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

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.

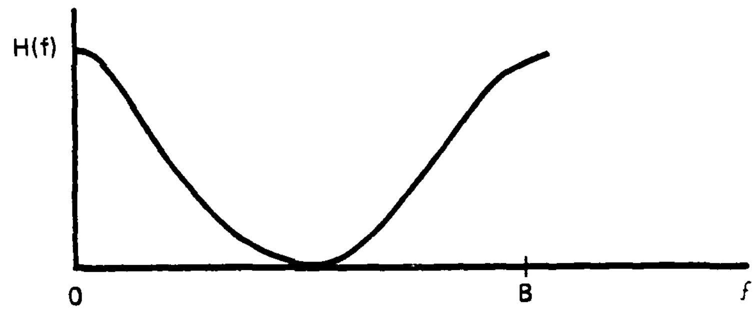
The time-averaged effect of beam jitter may be modeled as the convolution of the intensity profile with an appropriate probability density function (PDF) for the jitter statistics. The current SOQ model assumes that the jitter PDF is Gaussian with known mean and variance. The model allows the user to 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:

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

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.

$$\sigma = \theta_J \cdot Z \qquad \text{(A2)}$$

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 QLOT 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 specificed 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.

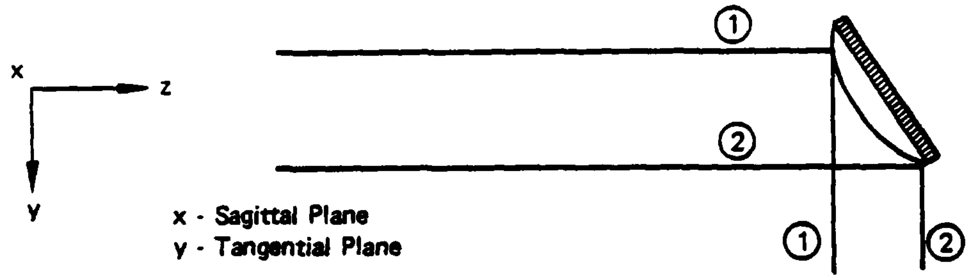
At the rear of this section are Fortran listings of the code updates which have been included in the basic Cycle III SOQ code previously documented.

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.



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

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

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

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

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

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

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

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

Similarly, the tangential plane is described by

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

$$S_T = \frac{-R_c \cos\phi}{2} \tag{A4}$$

$$= \frac{-R_c}{2\sqrt{2}} \text{ for } \phi = 45^\circ$$

The new mirror subroutine including astigmatic effects has the form

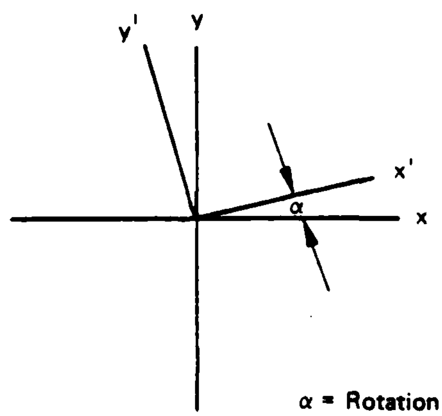
$$\Delta\theta_{SOQ} = \frac{2\pi}{\lambda} \left[\frac{x^2}{2S'_s} + \frac{y^2}{2S'_T} \right]$$

$$S'_s = \frac{R_c}{2 \cos \phi} \qquad S'_T = \frac{R_c \cos \phi}{2} \tag{A5}$$

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.



α = Rotation Angle

Since,

$$x = x \cos\alpha + y \sin\alpha \tag{A6}$$

$$y = -x \sin\alpha + y \cos\alpha \tag{A7}$$

Then,

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

Here,

(x,y) are the SOQ coordinates

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

The SOQ field is modified as

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

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
           XTOP(3,4),XSCR(4),AHC(14,20,9),TITLE3(20),XOPADD(4),
      *DELETE C10FLA.1
             DIMENSION IDIP(5,24),IGOL(99),AHC(14,20,9),CFFL(16384),IUSK(4,9),
      *DELETE       C10ASTG.2,C10ASTG.3
             DATA RANULS,DOUTY,DINY,PHIAST,PHIROT /5*0./
      *DELETE C10ASTG.4
           X DELTA,DISTF,DOUTY,DINY,RANULS,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,RANULS,RYOUT,RYIN,PHIAST,PHIRT,DESIPW)
```

```
*INSERT MIRROR.28
      PHIROT=PHIRT
*INSERT C10ASTG.12
        PHIROR=-PHIROT*PI/180.
      PHIROT=0.0
      WRITE(6,86)PHIROR
        SINPR=SIN(PHIROR)
        COSPR=COS(PHIROR)
*DELETE C10ASTG.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 C10ASTG.24
        XPRM=X(J)*COSPR+X(I)*SINPR
        YPRM=-X(J)*SINPR+X(I)*COSPR
        PHASE=AKY*((XPRM**2/RMSAG)+(YPRM**2/RMTAN))
*DELETE MIRROR.84
      PHIR=(PHIAST*PI)/180.
      RMSAG=ROC/COS(PHIR)
      RMTAN=ROC*COS(PHIR)
        PHIROR=-PHIROT*PI/180.
        PHIROT=0.0
        SINPR=SIN(PHIROR)
        COSPR=COS(PHIROR)
*DELETE MIRROR.91
        XPRM=X(J)*COSPR+X(I)*SINPR
        YPRM=-X(J)*SINPR+X(I)*COSPR
        PHIMIR=AKY*((XPRM**2/RMSAG)+(YPRM**2/RMTAN))
*INSERT MIRROR.108
      WRITE(6,86)PHIROR
      WRITE(6,420)RMSAG,RMTAN
 86     FORMAT(20X,*  MIRROR 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:

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

Momentum: $\rho\frac{D\vec{u}}{Dt} + \vec{\nabla}p = 0$

Energy: $\nabla\cdot\rho\vec{u}\ h + \frac{\vec{u}^2}{2} = q$

These are linearized assuming

$$\rho = \rho_\infty + \rho'$$

$$p = p_\infty + p'$$

$$\vec{u} = \hat{i}\,(U + u') + \hat{j}\,v' \qquad \text{(A10)}$$

Resulting in

$$\text{Continuity:} \quad \rho_\infty \, u'_x + \rho_\infty \, v'_y + U\rho'_x = 0 \qquad \text{(A11)}$$

$$\left(u'_x \equiv \frac{\delta}{\delta x}\, u'; \text{ etc}\right) \qquad \text{(A12)}$$

$$\text{Momentum:} \quad \rho_\infty \, \vec{U}u'_x + p_x = 0$$

$$\rho_\infty \, Uv'_x + p_y = 0 \qquad \text{(A13)}$$

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

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

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

$$v' = \frac{(\gamma - 1)q}{2\gamma \rho} \; \delta(x - \beta y) \qquad \text{(A16)}$$

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

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,

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

$$= -\frac{(\gamma-1)}{2\gamma p\beta}\int_{0}^{s} h(x=\beta y)\ \sin\mu ds \qquad \text{(A18)}$$

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

$$x = s\cos\mu \qquad y = s\sin\mu \qquad \text{(A19)}$$

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

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

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

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

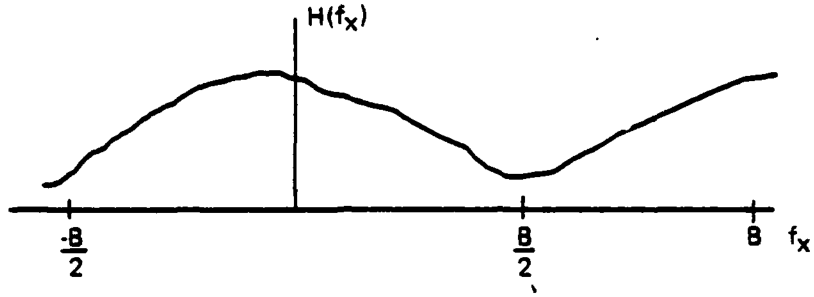
From sampling theory, the discrete values of f_x and f_y can be found since

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

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,

$$\frac{\Delta\rho}{\rho} = \frac{1}{\rho}\left[\frac{(\gamma - 1)M}{2a^3\beta}\int_0^s h(x,y)\Big|_{x=\beta y} \sin\mu \, ds\right.$$

$$\frac{(\gamma - 1)}{a^2 U}\iint dx'\,dy'\,h(x',y')\ \delta(y - y')\ I(x - x') \qquad \text{(A23)}$$

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.

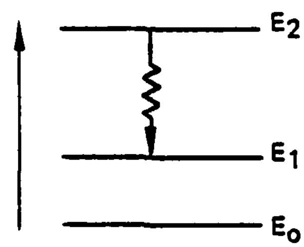
The heat release h(x,y) for a laser can be written:

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

where τ is the time constant for the depopulation of the lower laser level. If the depopulation were instantaneous ($\tau \to 0$), 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.

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

Where

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

the above expression can be inverted to give

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

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

$$\Delta H = c\Delta z \int_0^\infty dy \int_0^\infty dx \int_0^x \Delta I(x', y)\, e^{-(x - x')/U\tau}\, dx'$$

$$= c\Delta z \int_0^\infty dy \int_0^\infty dx \int_0^\infty I(x - x')\Delta I(x', y) e\ )^{-(x - x')/U\tau}\, dx' \tag{A27}$$

Where, recall

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

So,

$$\begin{aligned} \Delta H &= c\Delta z \int_0^\infty dy \int_0^\infty dx' \, \Delta I(x' y) \int_0^\infty dx \; I(x - x') e^{-(x - x')/U\tau} \\ &= c\Delta z \int_0^\infty dy \int_0^\infty dx' \, \Delta I(x', y) \int_{x'}^\infty dx'' e^{-x''/U\tau} \\ &= c\Delta z \int_0^\infty dy \int_0^\infty dx' \, \Delta I(x', y) \left(\frac{1}{1/U\tau}\right) \end{aligned} \quad \text{(A28)}$$

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:

$$\begin{aligned} w(x,y) &= \int_0^x dx' \, h(x', y) = c \int_0^x dx' \int_0^x dx'' \Delta I(x'', y) e^{-(x' - x'')/U\tau} \\ &= c \int_0^\infty dx' \, I(x - x') \int_0^\infty dx'' I(x' - x'') \Delta I(x'', y) e^{-(x' - x'')/U\tau} \\ &= c \int_0^\infty dx'' \Delta I(x'', y) \int_0^\infty dx' \, I(x - x') \; I(x' - x'') e^{-(x' - x'')/U\tau} \\ &= c \int_0^\infty dx'' \Delta I(x'', y) \; I(x - x'') \int_{x''}^x dx' \, e^{-(x - x'')/U\tau} \end{aligned} \quad \text{(A29)}$$

So,

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

So, recalling

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

And

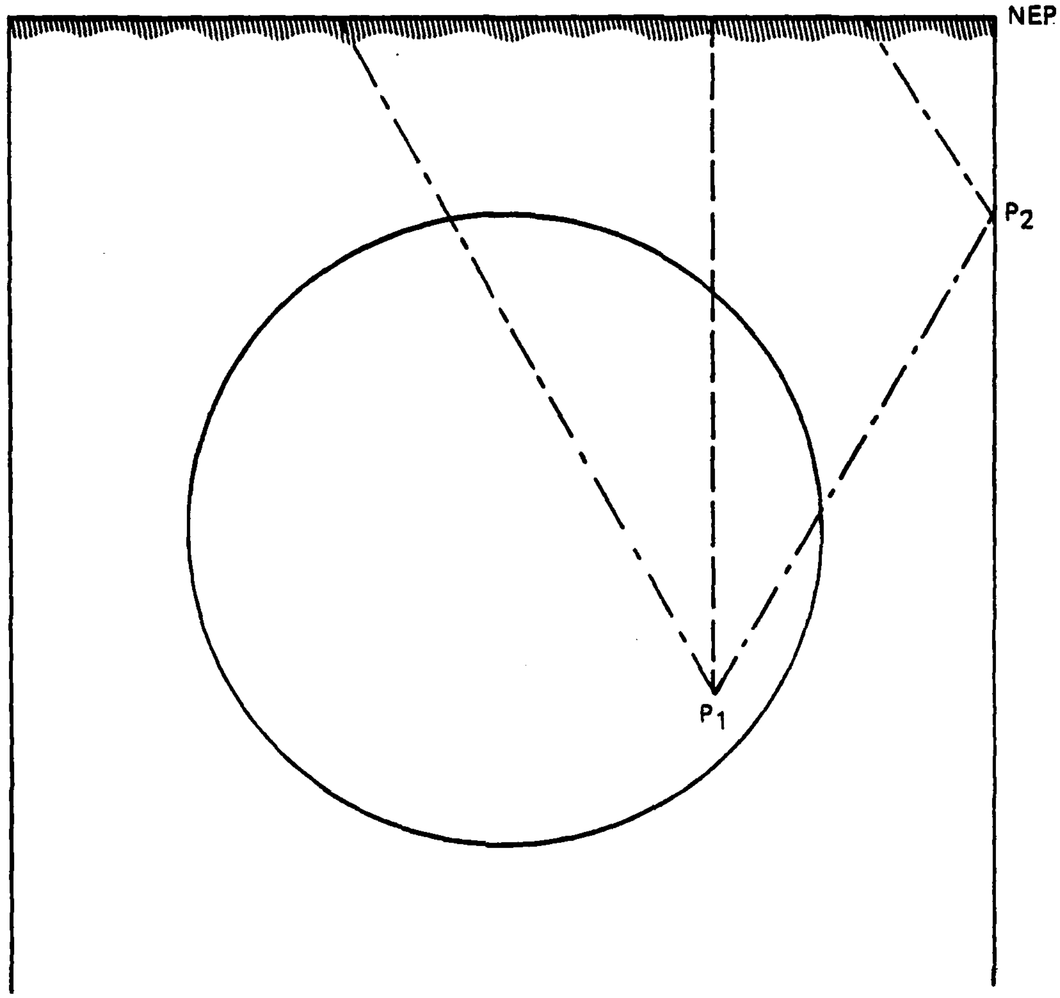
$$\Delta I(x'',y) = 2\left(\frac{1 - G}{1 + G}\right)PPD \qquad \text{from SIMPGG}$$

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

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:

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

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

$$n = 1 + C\rho \quad \text{(A34)}$$

The phase change $\Delta\phi$ is

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

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.

```
 :.F .T     HAHF

*IDENT BARE
      *INSERT GDL.531
           IF(IBARF.NE.1) GO TO 850
           RMIR = 1.0
           DELTA = 0.0
           DISTF = 0.0
           WRITE(6,860) IBARF
  860      FORMAT(/2X,10H**********,* IBARE =*,I2,* DELTA AND*,
          C * DISTF SET TO ZERO AND RMIR SET TO 1. *,10H**********/)
  850      CONTINUE
      *INSERT APR26.20
           WRITE(7) (CU(IZ),IZ=1,NOB)
           REWIND 7
      *DELETE          GDL.704,GDL.705
      *DELETE          GDL.690,GDL.702
           IF(ICAV.EQ.0) IBARE=1
           IF(IBARE.EQ.0) GO TO 691
           WRITE(29)(CFFL(IZ),IZ=1,NOB),X
           REWIND 29
           DO 689 I=1,NOB
  689        CU(I) = CFFL(I)
           DO 6891 I=1,NPTS
  6891       X(I) = XK(I)
           DIBMH  = ARC(1,1,1)/2.
           CALL APRTH(DIBMH,0.,0.,0.,0.,0.)
           POW=0.
           READ(29)(CFFL(IZ),IZ=1,NOB),X
           REWIND 29
           DO 688 I=1,NOB
  688        POW = POW + CU(I) * CONJG(CU(I))
           POW = POW *(XK(2) - XK(1))**2*(NPTS/NPY)
           WRITE(6,687) POW
  687        FORMAT(5X,*  ---- POWER IN FEEDBACK NORMED TO UNITY BY  *,
          XE15.7,/)
           SQTPOW  = SQRT(POW)
           DO 686 I=1,NOB
  686        CFFL(I) = CFFL(I) / SQTPOW
      *DELETE          GDL.822,GDL.823
           IF(IBARE.EQ.0) GO TO 1002
           IF(INIT.AND..NOT.RESTRT)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 EST),*, /)
           GO TO 1003
```

```
 1002 CONTINUE
      CALL REGAIN(NCT,NITER)
 1003 IF(ICEK.EQ.0) GO TO 565
*DELETE          GDL.828,GDL.838
*DELETE CAVITY.3
     X ZLI,ZLO,IBARE)
*INSERT CAVITY.141
      IF(IBARE.NE.0) GO TO 109
*DELETE CAVITY.183
  109 WRITE (7) (CU(IZ),IZ=1,MUT)
*DELETE CORR1.47
      CALL DENSY(FLAG,RHO,XLEN,YLEN,UCZ,NXA,NYA,1,IN,NNSYM,IBARE)
*INSERT SO077CY1.10
      IF(IBARE.GT.0) GO TO 12
*INSERT CAVITY.217
      GO TO 11
   12 CG(IZ)=CMPLX(COS(PHIM),SIN(PHIM))
*DELETE DENSY.2
      SUBROUTINE DENSY(FLAG,RHO,XLEN,YLEN,ZSLAB,NPX,NPY,IF,IN,NSYM,
     X IBARE)
*INSERT DENSY.105
      IF(IBARE.EQ.1) RETURN
      IF(LAG.GT.0) GO TO 12
      WRITE(6,13)
   13 FORMAT(//10X,5H***  ,*FLAG = 0. IN DENSY*,5H  ***/
     A  15X,              *ALL JPDS SET TO 0.0*//)
      RETURN
   12 CONTINUE
*INSERT LROP1.2
      DATA IBARE/0/
*INSERT GDL.47
      DATA DESIPW/0./
*INSERT LROP1.1
      COMMON /BARES/ IBARE
*INSERT GDL.22
      COMMON /BARES/ IBARE
*DELETE LROP1.3
     X ,IBARE,PLOTS
C
C      IBARE IS FLAG FOR LOADED, BARE, OR SEMI-BARE CAVITY
C            = 0 FOR LOADED RESONATOR (DEFAULT VALUE)
C            = 1 FOR BARE RESONATOR (UNITY GAIN,0 PHASE CHANGE)
C            = 2 FOR SEMI-BARE RESONATOR (UNITY GAIN,DENSITY PHASE CHANGE)
*INSERT GDL.430
       IDIR(5,ICAV) = IBARE
*INSERT GDL.437
     X,IDIR(5,ICAV)
*DELETE GDL.439
     X   PROPAGATING PARAMETER ,I2/* IBARE= *,I3)
*DELETE GDL.444
     X RESTRT,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
       NOB = NPTS*NPY
*DELETE MIRROR.24
   70 IF(DESIPW.EQ.0.0) GO TO 360
       NOB = NPY*NPTS
       NOB2 = NOB*2
C *** FIND INCIDENT POWER
```

IDENT HAWE

```
      PPWIN = 0.0
      DO 355 IZ=1,NOB2,2
355   PPWIN = PPWIN + CUR(IZ)**2 + CUR(IZ+1)**2
      PPWK = PPWIN*(X(2)-X(1))**2*(NPTS/NPY)/1000.
      IF(NREG.EQ.1.OR.NREG.EQ.2) PPWK=PPWK/WNOW**2
      TRANS = SQRT(DESIPW/PPWK)
C **** SCALE THE BEAM TO THE DESIRED POWER.
        DO 356 IZ = 1,NOB2
356     CUR(IZ) = CUR(IZ)*TRANS
        WRITE(6,4010) DESIPW,PPWK,TRANS
4010    FORMAT(/5X,42HTHE FIELD HAS BEEN SCALED TO DESIRED POWER/
     X  8X,12HDESIPW      =,G12.4/8X,12HPPWK        =,G12.4/
     X  8X,12HTRANS       =,G12.4/)
      DESIPW = TRANS
  360 IF(ABS(ANX).LE.0.000100.AND.ABS(ANY).LE.0.000100) GO TO 71
*DELETE C10ASTG.7
     3 ABC(1,IMIR,4),ABC(12,IMIR,2),ABC(13,IMIR,2),DESIPW)
      IF(DESIPW.EQ.0.0) GO TO 24
C  ****** SCALE THE FIELD BACK DOWN.
      DO 358 IZ=1,NO9
358     CU(IZ) = CU(IZ)/DESIPW
        WRITE(6,4000)
4000   FORMAT(/5X,30HTHE FIELD HAS BEEN SCALED DOWN/)
24     CONTINUE
```

IDENT JITTER

```
*IDENT JITTER
*DELETE GDL.20
     X ICUT,MLT,IDK,IJTR,
     X ICNT24,ICNT25,ICNT26,
     X ITM,ICEK,NCT
*DELETE GDL.26
     XDSMM(20),RMV(20),PHIA(20),RCURVE(4),DSP(4),TLT(4),ICAVZ(20)
*DELETE GDL.260
      DO 173 IZERO=1,20
*DELETE GDL.314
      DO 177 IZERO=1,17
*INSERT S0077CY1.165
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/
*DELETE GDL.295,S0077CY1.167
C    / 16/ 17/ 18/ 19/ 20/ 21/ 22/ 23/ 24/ 25/ 26/
     X,160,170,180,190,200,210,365,230,240,250,260),IFLOW
*DELETE GDL.325,S0077CY1.168
C    / 16/ 17/ 18/ 19/ 20/ 21/ 22/ 23/ 24/ 25/ 26/
     X,160,170,180,190,200,210,365,230,240,250,260),IFLOW
*INSERT GDL.243
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 GOL.327
C   ********************************************************
C
  230 IJTR = IJTR+1
      IF(.NOT.INIT) GO TO 231
      READ(IN,JITTER)
      ABC(6,IJTR,1) = JITANG*1.E-6
      ABC(6,IJTR,2) = JITDIS
  231 SIGXY=ABC(6,IJTR,1)*ABC(6,IJTR,2)
      WRITE (6,1836) ABC(6,IJTR,1),SIGXY
 1836 FORMAT (45H **** BEAM JITTER MODEL CALLED ****,STD DEVIA,
     X23HTION ANGLE (RADIANS) = ,G12.5,8X,*: STD DEVIATION (SIGXY) =*,
     XG12.5)
      DO 233 IJ=1,NOB
  233 US(IJ) = CUR(2*IJ-1)**2 + CUR(2*IJ)**2
      DX = X(2) - X(1)
      CALL JITRBG(DX,SIGXY)
      DO 235  IJ=1,NOB
      IJIJ = 2*IJ
      CUR(IJIJ-1) =SQRT(US(IJ))
  235 CUR(IJIJ) = 0.0
  240 CONTINUE
  250 CONTINUE

  260 CONTINUE
      IGNAL = 1
      GO TO 999
*INSERT GOL.32
      EQUIVALENCE (US(1),CFIL(1)),(CUR(1),CU(1))
      DIMENSION US(16384),CUR(32768)
*INSERT GOL.15
     X ,US,CUR
      REAL JITANG,JITDIS
*INSERT CYCLE9.67
      DO 134 II=1,NPTS
  134 X(II) = XSAVE(II)
*DELETE QUAL.2
      SUBROUTINE QUAL (IPHASE,ISAVE,IPLT,TITLE,RH,ANS,DR,RF,SIGANG)
*INSERT QUAL.11
      DATA PI/3.14159266/
*DELETE QUAL.107
   63 IF(SIGANG-1.E-9)70,70,66
   66 SIGXY=F*SIGANG
      WRITE (6,1836) SIGANG
 1836 FORMAT (45H **** BEAM JITTER MODEL CALLED ****,STD DEVIA,
     X14HTION ANGLE =  , G20.5 )
      CALL JITRBG(DXSAVE,SIGXY)
      UMAX=0.
      DO 68 J=1,NPTS
      JI=(J-1)*NPTS
      DO 68 I=1,NPTS
      IZ=I+JI
      IF (UMAX.GE.US(IZ))GO TO 68
      UMAX=US(IZ)
      XPEAK=X(I)
      YPEAK=X(J)
      IZIZ=2*IZ
      CUR(IZIZ-1) =SQRT(US(IZ))
      CUR(IZIZ) = 0.0
   68 CONTINUE
   70 UMXR=UMAX/1000.
*INSERT MAIN.300
      SUBROUTINE JITRBG(DX,SIGXY)
C     THIS SUBROUTINE MODIFIES THE FAR FIELD INTENSITY DISTRIBUTION
```

```
C      MODEL THE EFFECTS OF BEAM JITTER.  THE JITTER IS ASSUMMED TO
C      GAUSSIAN.  SINCE THE RESULTING INTENSITY IS THE CONVOLUTION OF THE
C      UNDISTURBED INTENSITY WITH THE GAUSSIAN. THE OPERATION IS PERFORME
C      BY THE FFT ON EACH FUNCTION ALONE. MULTIPLYING THE RESULTS.
C      PERFORMING THE INVERSE FFT. JVF.6/24/76.
       LEVEL 2.CU.CI
       COMPLEX CU.CI
       COMMON /MELT/CU(16384).US(33024).X(128).WL.NPTS.NPY.DRX.DRY
       COMMON /CG/CI(17100)
       DIMENSION NND(2)
       DATA PI /3.141593/
       NPTSQ = NPTS * NPTS
       PPWN = 0.0
       PPW = 0.0
       DO 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.*(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 JITRBG.*//)
          RETURN
          END
*DELETE MAIN.60
          NAMELIST/QLOT/TITLE.IQLT.DB.ISAV.IPHASE.RBB.RF.SIGANG
*DELETE MAIN.230
   210 CALL QUAL (IPHASE.ISAV.IQLT.TITLE.RBB.AS.DB.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	$70R^8 - 140R^6 + 560R^6 - 210R^4$
12	$(4R^4 - 3R^2)\cos 2\theta$	24	$252R^{10} - 630R^8 + 560R^6 - 210R^4 + 30R^2 - 1$

The phase applied is

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

$$= \Delta\phi(1,\theta) \qquad R>1 \tag{B1}$$

If the Zernike radius R_o 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_o 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.

```
IDENT  ZRNIKE

*IDENT ZRNIKE
 GDL
     *DELETE JITTER.55
           * IZERN,ICNT24,ICNT24,
     *DELETE JITTER.62
     C            = 24  APPLY UP TO 24 ZERNIKES IN UNITS OF WAVES
     *DELETE JITTER.78
     *INSERT JITTER.83
           LOGICAL FRINGE,REVOJ,REVOJS,RWIND,RWINDS,RGAIN
     *INSERT JITTER.102
     C**************************************************************
     C      APPLY ZERNIKE
     C**************************************************************
       240 IZERN = IZERN + 1
           IF (.NOT.IZ(I)) GO TO 244
           FRINGE = .FALSE.
           DO 248 I=1,24
       248 P(I) = 0.
           DO 244 I=1,35
       244 PFRNG(I) = 0.
           READ (5,ZERNS)
           DO 233 I=1,35
       233 IF(PFRNG(I).NE.0.) FRINGE=.T.
           IF(.NOT.FRINGE) GO TO 261
           WRITE(6,245)
       245 FORMAT(/5X,*FRINGE COEFFICIENTS BEING CONVERTED TO SOQ ORDER.*/)
           P(1) = 0.
           P(2) = PFRNG(1)
           P(3) = PFRNG(2)
           P(4) = PFRNG(3)
           P(5) = PFRNG(4)
           P(6) = PFRNG(5)
           P(7) = PFRNG(6)
           P(8) = PFRNG(7)
           P(9) = PFRNG(8)
           P(10) = PFRNG(10)
           P(11) = PFRNG(9)
           P(12) = PFRNG(11)
           P(13) = PFRNG(12)
           P(14) = PFRNG(16)
           P(15) = PFRNG(17)
           P(16) = PFRNG(13)
```

```
      R(17) = RFRNG(14)
      R(18) = RFRNG(18)
      R(19) = RFRNG(13)
      R(20) = RFRNG(25)
      R(21) = RFRNG(26)
      R(22) = RFRNG(15)
      R(23) = RFRNG(24)
      R(24) = RFRNG(35)
      IFRIST = 0
      DO 246 K=20,23
  246 IF(RFRNG(K).NE.0.) IFRIST = 1
      DO 243 K=27,34
  243 IF(RFRNG(K).NE.0.) IFRIST = 1

      IF(IFRIST.EQ.1) WRITE(6,247)
  247 FORMAT(/5X,*WARNING - FRINGE COEFFICIENTS OF ORDER 20 THROUGH 23*,
     C * AND 27 THROUGH 34 ARE IGNORED*/)
  241 DO 242 I=1,24
  242 RZSAVE(I,IZERN) = R(I)
      RZSAVE(25,IZERN) = R0
  244 CALL ZERN(RZSAVE(25,IZERN),RZSAVE(1,IZERN))
      IGNAL = 1
      GO TO 999
*DELETE GOL.27
      DIMENSION IPLTS(50),RZSAVE(25,10),R(24),IIPLT(20),
     X RFRNG(35),READ35(9),RWINDS(9),OTITL(9,20)
*INSERT GOL.33
      DATA R,RFRNG/24*0.,35*0./ , R0 / 5. /
*INSERT GOL.243
C
      NAMELIST /ZERNS/ R0,R,RFRNG
C
C           R0 = ZERNIKE NORMALIZATION RADIUS.
C           R  = ARRAY OF ZERNIKE COEFFICIENTS TO BE APPLIED.
C           RFRNG = ARRAY OF FRINGE ZERNIKE COEFFICIENTS TO BE APPLIED.
TILT
*INSERT LPROP1.345
      SUBROUTINE ZERN(R0,R)
      LEVEL 2,CUR
      COMMON /MELT/ CUR(32768),CFIL(16512),X(128),WL,NPTS,NPY,DRX,DRY
      COMPLEX CFIL
      DIMENSION R(24)
      IF(R0.EQ.0.) GO TO 70
      DO 100 IY=1,NPY
      J1 = (IY-1)*NPTS
      YSQ = X(IY)**2
      DO 100 IX=1,NPTS
      XSQ = X(IX)**2
      INDX = IX + J1
      R = SQRT(XSQ+YSQ)
      THET = ATAN2(X(IY),X(IX))
      R = AMIN1(R/R0,1.)
      CT = COS(THET)
      C2T = COS(2.*THET)
      C3T = COS(3.*THET)
      C4T = COS(4.*THET)
      C5T = COS(5.*THET)
      ST = SIN(THET)
      S2T = SIN(2.*THET)
      S3T = SIN(3.*THET)
      S4T = SIN(4.*THET)
      S5T = SIN(5.*THET)
      R2  = R**2
      R3 = R*R2
      R4 = R*R3
      R5 = R*R4
      R6 = R*R5
```

```
      R8 = R2*R6
      R10 = R2*R8
      DEL = P(1)   + P(2)*R*CT   + P(3)*R*ST
     A      + P(4)*(2.*R2-1.)
     B      + P(5)*R2*C2T     + P(6)*R2*S2T
     C      + P(7)*(3.*R3-2.*R)*CT   + P(8)*(3.*R3-2.*R)*ST
     D      + P(9)*R3*C3T     + P(10)*R3*S3T
     E      + P(11)*(6.*R4-6.*R2+1.)
     F      + P(12)*(4.*R4-3.*R2)*C2T   + P(13)*(4.*R4-3.*R2)*S2T
     G      + P(14)*R4*C4T     +    P(15)*R4*S4T
     H      + P(16)*(10.*R5-12.*R3+3.*R)*CT
     I      + P(17)*(10.*R5-12.*R3+3.*R)*ST
     J      + P(18)*(5.*R5-4.*R3)*C3T + P(19)*(5.*R5-4.*R3)*S3T
     K      + P(20)*R5*C5T     +    P(21)*R5*S5T
     L      + P(22)*(20.*R6-30.*R4+12.*R2-1.)
     M      + P(23)*(70.*R8-140.*R6+90.*R4-20.*R2+1.)
     N      + P(24)*(252.*R10-630.*R8+560.*R6-210.*R4+30.*R2-1.)
      INO2 = IOOI*2
      DEL  = DEL*2.*3.141592654
      COSD = COS(DEL)
      SIND = SIN(DEL)
      CORS = COR(INO2-1)
      COR(INO2-1) = CORS*COSD - COR(INO2)*SIND
  100 COR(INO2)   = CORS*SIND + COR(INO2)*COSD
      WRITE (6,200) RHO
  200 FORMAT (*0ZERNIKE PHASE CORRECTION APPLIED WITH NORMALIZATION*
     A * RADIUS OF *,E15.4  /* COEFFICIENTS USED P(1)-P(24)*,
     B * ARE CONSISTENT WITH THE PHASE DUE TO THE NTH TERM BEING*//
     C 20X,*PHI(N) = 2*PI*P(N)*Z(N)//
     D * Z(N) = RF(R)*.1H*.*F(THETA)( RF(N) NORMALIZED TO 1. AT R=1.*//
     E  (1X,5E20.5))
      RETURN
   70 NOR = NPTS*NPY
      DO 40 I=1,NOR
      II=I+I
      IIM1=II-1
      COR(IIM1) = SQRT(COR(II)**2+COR(IIM1)**2)
   40 COR(II) = 0.0
      WRITE(6,300)
  300 FORMAT(//10X,*CO 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.

FORTRAN:

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*SECOND(T)
   REGAIN
      *DELETE REGAIN.93
             DELT = (IFIN-ISRT)/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.

FORTRAN:

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 READ(K.5) IC1.C
             IF (EOF(K).NE.0.0) GO TO 15
           5 FORMAT(I1.20A4)
             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(IRSDS.NE.0) CALL LISTER(IRSDS)
      IF(NWL.LE.0.) REWIND 50
      IRSDS = 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 TO LEVEL 2 ARE FOLLOWED BY "(2)".
   C
   C     ROUTINE        COMMON          EXTERNAL              CALLED BY
   C     -----------------------------------------------------------------
   C      SOQ           FST  (2)        CNSTRN
   C                    MELT  (2)       DAVIDN
   C                    PLTSIG          GOL
   C                    INITL           ISOS
   C                    GLAD            LISTER
   C                    SVTYM           LISTRD
   C                    NAMES           NEAR
   C                                    PRETYP
   C                                    QUAL
   C      JITRAG        MELT  (2),      FOURT                 GOL
   C                    CG  (2)                               QUAL
   C      ISO           -               -                     -
   C      ERFC          -               ERF                   GAINXY
   C                                                          KINET
   C                                                          THERML
```

```
C     ISOCAV      -           -                REGAIN
C     ISOS        -           -                SOQ
C     PRETYP      -           -                SOQ
C     THREED      -           -                NEAR
C     VINO        -           -                REGAIN
C     CPUTIM      -           -                FUHS
C                                              GAINXY
C                                              GOL
C                                              REGAIN
C     LISTER      -           -                SOQ
C     LISTRO      -           -                SOQ
C     DATE        -           -                NEAR
C     HCLOCK      -           -                NEAR
C     AEROW       MELT (2)    -                GOL
C     RANDU       -           -                -
C     APRTR       MELT (2)    -                FIELDS
C                 WAY                          GOL
C                                              MIRROR
C     AXICN       MELT (2)    INTERP           GOL
C                             SPTAN
C     BLIMIT      CAV2 (2)    -                REGAIN
C     CAVITY      REACTR      DENSY            GOL
C                 WAY         GAINXY
C                 CAV2 (2)    INTERP
C                 MELT (2)    OUTPUT
C                 CCG (2)     STEP
C                 GLAD
C                 CAVX (2)
C                 ARPROP
C                 SEGCV2
C                 STROWL
C     CEMBAR      -           -                QUAL
C     CNSTRM      -           -                SOQ
C     CONEFF      MELT (2)    -                NEAR
C     DAVTIN      -           -                SOQ
C     DENSY       MELT (2)    LINTERP          CAVITY
C                 LENSY       ROSS
C                 GLAD        ROSNA
C                 SEGDEN
C     FIELDS      MELT (2)    APRTR            GOL
C     FOURT       -           -                JITHAG
C                                              RSTEP
C                                              STEP
C     FUHS        CAV2 (2)    CPUTIM           REGAIN
C                 GLAD
C     GAINXY      START       CPUTIM           CAVITY
C                 REACTR      FREC             REGAIN
C                 PROPT       KINET
C                 MOLES       MIX
C                 ENERG
C                 RATE
C                 FACTER
C                 CAV2 (2)
C                 GLAD
C     GOL         MELT (2)    AEROW            SOQ
C                 ARPROP      APRTR
C                 WAY         AXICN
C                 ZIP         CAVITY
C                 DAZ         CPUTIM
```

```
C
C                 INITL          FIELDS
C                 SEGGOL         INTERP
C                 STRGWL         IPLOT
C                 SVTYM          JITRRG
C                 WARES          MIRROR
C                                POWR
C                                REGAIN
C                                RGRD
C                                RSTEP
C                                SLIVER
C                                SPIDER
C                                STEP
C                                TALOOM
C                                THERML
C                                ZERN
C     INTERP      -              -                  AXICN
C                                                   CAVITY
C                                                   GDL
C     IPLOT       MELT (2)       OUTPUR             GDL
C                 WAY            OUTPUT             QUAL
C                 PLTSIG
C     KINET       PROPT          EREC               GAINXY
C                 START          MIX
C                 MOLES
C                 ENERG

C                 RATE
C                 FACTER
C                 GFACTR
C     MIRROR      MELT (2)       APRTR              GDL
C                 MIRROR
C                 WAY
C     MIX         PROPT          -                  GAINXY
C                 MOLES                             KINET
C                 RATE
C     MODER       MELT (2)       SPTAN              -
C                 WAY
C     NEAR        MELT (2)       CONEFF             SOQ
C                 TIME           DATE
C                 VIEW           HCLOCK
C                                THREED
C     OUTPUR      WAY            -                  IPLOT
C     OUTPUT      WAY            -                  CAVITY
C                 PLTSIG                            IPLOT
C                                                   TALOOM
C                                                   THERML
C     PLTOT       -              POWWOW             QUAL
C     POWR        -              -                  GDL
C     POWWOW      -              -                  PLTOT
C                                                   QUAL
C     QUAL        MELT (2)       CENBAR             SOQ
C                 STRGWL         JITRRG
C                                IPLOT
C                                PLTOT
C                                POWWOW
C                                STEP
C                                TILT
C     REGAIN      MELT (2)       ALUMIT             GDL
```

```
C                    CCG (2)          CPUTIM
C                    CAV2 (2)         FUHS
C                    GLAD             GAINXY
C                                     ISOCAV
C                                     SIMPGG
C                                     VINO
C        RGRD        MELT (2)         -                     GDL
C        ROSN        LENSY            -                     DENSY
C        LINTERP     LENSY            -                     DENSY
C        ROSN6       MELT (2)         -                     DENSY
C        RSTEP       WAY              FOURT                 GDL
C                    MELT (2)
C        SIMPGG      CAV2 (2)         -                     REGAIN
C                    GFACTR
C        SLIVER      MELT (2)         -                     GDL
C        SPIDER      MELT (2)         -                     GDL
C        SPIAN       -                -                     AXICN
C                                                           MODER
C        STEP        WAY              FOURT                 CAVITY
C                    MELT (2)         TILT                  GDL
C                    STPLCM (2)                             QUAL
C                    STPAWL                                 TRLOOM
C        TRLOOM      MELT (2)         OUTPUT                GDL

C                    WAY              STEP
C        THERML      MELT (2)         FREC                  GDL
C                    WAY              OUTPUT
C        TILT        MELT (2)         -                     QUAL
C                                                           STEP
C        ERF         -                -                     ERFC
C        ZERN        MELT (2)         -                     GDL
C     --------------------------------------------------------------
C
C
C
C     COMMON        ROUTINES
C     --------------------------------------------------------------
C        BARES       SOQ.GDL
C        CAVX (2)    CAVITY
C        CAV2 (2)    ALUMIT.CAVITY.FUHS.GAINXY.REGAIN.SIMPGG
C        CCG (2)     CAVITY.REGAIN
C        CG (2)      JITRHG
C        ENERG       GAINXY.KINET
C        FACTER      GAINXY.KINET
C        FST (2)     SOQ
C        GFACTR      CAVITY.GAINXY.KINET.SIMPGG
C        GLAD        SOQ.CAVITY.DENSY.FUHS.GAINXY.REGAIN
C        INITL       SOQ.GDL
C        LENSY       DENSY.ROSN.LINTERP
C        MELT (2)    SOQ.JITRHG.AFROW.APRTR.AXICON.CAVITY.CONFFE.
C                    DENSY.FIELDS.GDL.IPLOT.MIRROR.MODER.NFAR.QUAL.
C                    REGAIN.RGRD.ROSN6.RSTEP.SLIVER.SPIDER.STEP.
C                    TRLOOM.THERML.TILT.ZERN
C        MOLES       GAINXY.KINET.MIX
C        MIRROR      CAVITY.GDL.MIRROR
C        PLTSIG      SOQ.IPLOT.OUTPUT
C        PROPT       GAINXY.KINET.MIX
C        JAZ         GDL
```

```
C       HALF          GAINXY,KINET,MIX
C       SEGCV2        CAVITY
C       SEGDEN        DENSY
C       SEGGOL        GOL
C       START         GAINXY,KINET
C       STPLCM  (2)   STEP
C       STPOWL        CAVITY,GOL,OUAL,STEP
C       SVTYM         SOQ,GOL
C       TIME          NEAR
C       VIEW          NEAR
C       WAY           APRTR,CAVITY,GOL,IPLOT,MIRROR,MODER,OUTPUR,
C                     OUTPUT,RSTEP,STEP,TBLOOM,THERML
C       ZIP           GOL
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(8,1,1)        :    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(8,ISTEP,3)    :    IFLOW=3, PROP
     C      ABC(1,IAP,4)      THROUGH     ABC(8,IAP,4)      :    IFLOW=4, APRTR
     C      ABC(1,IDK,5)      THROUGH     ABC(10,IDK,5)     :    IFLOW=5, TBLOOM
     C      ABC(1,IJTR,6)     THROUGH     ABC(2,IJTR,6)     :    IFLOW=23, JITTER
     C      ABC(10,IWFA,6)                                  :    IFLOW=15, REGRID
     C      ABC(1,ITHRML,7)   THROUGH     ABC(8,ITHRML,7)   :    IFLOW=17, THERML
     C      ABC(1,IRSTP,8)    THROUGH     ABC(4,IRSTP,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:

a. 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  PLTFIX

  *IDENT PLTFIX
   IPLOT
      *INSERT LROP1.64
            DCF = 0.
      *INSERT LROP1.66
            DCF = DCF + US(J)
      *INSERT LROP1.67
            DCF = DCF*(X(2)-X(1))**2*FLOAT(NPTS/NPY)/1000.
      *DELETE LROP1.86,LROP1.87
         38 WRITE(6,6) ADIM,UMAXR,DCF,DRX,DRY
          6 FORMAT(12H    DCALC = ,G11.5,4X,7HIMAX = ,G11.4,4X,6HDCALC ,
           X   7HFLUX = ,G11.5//24X,
      *DELETE LROP1.96
       1500 IF(XAXIS) WRITE(6,786)
   GDL
      *DELETE GDL.158
            NAMELIST /PLOT/ KPLOT,TITLE,RADPLT
      C            KPLOT = ABCDE, WHERE A,B,C,D,AND E ARE 0 OR 1.
      C                  A = 1         RADIAL PLOTS
      C                  B = 1         ISOINTENSITY PLOTS
      C                  C = 1         X - AXIS PLOTS
      C                  D = 1         DIAGONAL PLOTS
      C                  E = 1         Y - AXIS PLOTS
      *INSERT GDL.503
            KPLOT = 0
      *INSERT GDL.505
            IF(RADPLT.NE.0..AND.KPLOT.EQ.0) KPLOT=11111
            IF(RADPLT.EQ.0..AND.KPLOT.EQ.0) KPLOT=1111
            IIPLT(IPTT) = KPLOT
      *DELETE GDL.510,GDL.511
            KPLOT = IIPLT(IPTT)
            CALL IPLOT(KPLOT)
   IPLOT
      *INSERT LROP1.30
            DIMENSION B(14),XII(128)
      *INSERT LROP1.35
            DATA BLANK,DOT /1H ,1H./
            DATA B /1H ,1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H*,1H-,1HI/
      *DELETE LROP1.76,LROP1.78
            U1 = US(I2)/UMAX
            IK = 10.*U1 + 2
            IF(U1.LT..01) IK=1
            IF(I.EQ.1.OR.I.EQ.NPY) IK=14
            IF(J.EQ.1.OR.J.EQ.NPTS) IK=13
          2 XII(I) = B(IK)
          4 WRITE(6,3) (XII(I),I=1,NPY)
          3 FORMAT(1X,128A1)
```

```
*DELETE LROP1.43,LROP1.45
      U1 = US(I2)/UMAX
      IK = 10.*U1 + 2
      IF(U1.LT..01) IK=1
      IF(I.EQ.1.OR.I.EQ.NRY) IK=14
      IF(J.EQ.1.OR.J.EQ.NRTS) IK=13
   12 XIT(I) = H(IK)
   14 WRITE(6,13) X(J),(XIT(I),I=1,NRY)

   13 FORMAT(1X,F10.2,2X,64A1)
```

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

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

CAVITY
  *INSERT CAVITY.294
        POWA = 0.
        POWB = 0.
```

```
*DELETE CAVITY.286
      MXMX = 2*MX
      XINT = (CUR(MXMX)**2 +CUR(MXMX-1)**2)*XFACT
      POWB = POWB + XINT
   61 US(MX) = XINT
      POWB = (POWB*(X(2)-X(1))**2*FLOAT(NPTS/NPY))/1000.
*DELETE CAVITY.323
      JYJY = 2*JY
      XINT = (CUR(JYJY)**2 +CUR(JYJY-1)**2)*XFACT
      POWA = POWA + XINT
   64 US(JY) = XINT + US(JY)
      POWA = (POWA*(X(2)-X(1))**2*FLOAT(NPTS/NPY))/1000.
      WRITE(6,62) JNS,NCAVN,POWB,POWA
   62 FORMAT(//3X,18HGAIN/PHASE SEGMENT,I2,17H IN CAVITY NUMBER,I2,
     X28H HAS BEEN APPLIED. FLUX IN =,G14.7,13H.  FLUX OUT =,G14.7/)
STEP
*INSERT STEP.14
      DATA NREG,WNOW /0,1.0/
      IF(IPRNT.NE.0) WRITE(6,1000) DXREAL,DYREAL,RAPTR,NREG,WNOW
 1000 FORMAT(5X,*ENTERING SUBROUTINE STEP. CURRENT PROPAGATION*,
     X * PARAMETERS:*,
     X /8X,*DRX          =*,G12.4,
     X /8X,*DRY          =*,G12.4,
     X /8X,*RAPTR        =*,G12.4,
     X /8X,*NREG         =*,I8,
     X /8X,*WNOW         =*,G12.4)
      POWB = 0.
      POWA = 0.
      NOH = NPTS*NPY
      DO 400 I=1,NOH
      II = 2*I
  400 POWB = POWB + CUR(II-1)**2 + CUR(II)**2
      POWB = POWB*(X(2)-X(1))**2*FLOAT(NPTS/NPY)/1000.
      IF(NREG.EQ.1.OR.NREG.EQ.2) POWB = POWB/WNOW**2
*INSERT STEP.234
      NOH = NPTS*NPY
      DO 401 I=1,NOH
      II = 2*I
  401 POWA = POWA + CUR(II-1)**2 + CUR(II)**2
      POWA = POWA*(X(2)-X(1))**2*FLOAT(NPTS/NPY)/1000.
      IF(NREG.EQ.1.OR.NREG.EQ.2) POWA = POWA/WNOW**2
      DELP = (POWB-POWA)/POWB*100.
      IF((ITH.EQ.0.OR.NREG.EQ.0).AND.IPRNT.NE.0) WRITE(6,3000)
     X  DXREAL,DYREAL,NREG,WNOW,DELP
 3000 FORMAT(/5X,*EXITING SUBROUTINE STEP. CURRENT PROPAGATION*,
     X * PARAMETERS:*,

     X /8X,*DRX                    =*,G12.4,
     X /8X,*DRY                    =*,G12.4,
     X /8X,*NREG                   =*,I8,
     X /8X,*WNOW                   =*,G12.4,
     X /8X,*PERCENT FLUX LOST      =*,G12.4)
*DELETE STEP.191
   16 FORMAT(/4X,19H STREHL INTENSITY =,G12.5)
*INSERT STEP.255
      WNOW = 1.0
      IF(IPRNT.NE.0) WRITE(6,3000) DXREAL,DYREAL,NREG,WNOW,DELP
*INSERT STEP.264
      WNOW = 1.0
GOL
```

```
*DELETE GDL.847,GDL.848
      UMAX = 0.0
      XMAX = X(1)
      YMAX = X(1)
      DO 78 J=1,NPY
      J1 = (J-1)*NPTS
      DO 78 I=1,NPTS
      IZ = I+J1
      XYINT = CU(IZ)*CONJG(CU(IZ))
      IF(UMAX.GT.XYINT) GO TO 78
      UMAX = XYINT
      XMAX = X(I)
      YMAX = X(J)
78    PPW = PPW+XYINT
      IF(NREG.EQ.1.OR.NREG.EQ.2) UMAX=UMAX/WNOW**2
      UMAXK = UMAX/1000.
      RADMAX = SQRT(XMAX**2+YMAX**2)
*DELETE GDL.868
      IF(MSTEP.NE.1) WRITE(6,79) PPWK,DCALCP,UMAXK,XMAX,YMAX,RADMAX
*DELETE GDL.870
     XUX =,G12.4/8X,12HDCALC      =,F8.2/8X,12HIMAX       =,G12.4,10X,
     X20HLOCATED AT (X,Y) = (,G12.4,1H,,G12.4,1H),
     X /42X,9HAT RADIUS,G12.5)
*DELETE GDL.871
      IF(MSTEP.EQ.1) WRITE(6,779) PPWK,DCALCP,UMAXK,XMAX,YMAX,RADMAX
*DELETE GDL.873
     XUX =,G12.4/8X,12HDCALC      =,F8.2/8X,12HIMAX       =,G12.4,10X,
     X20HLOCATED AT (X,Y) = (,G12.4,1H,,G12.4,1H),
     X /42X,9HAT RADIUS,G12.5)
*DELETE GDL.512,GDL.513
      IGOAL = 1
      GO TO 999
TILT
*DELETE CYCLE9.233
      TWORDC = 2.*RADCUR
      IF(IPS.GE.2) WRITE(6,67) RADCUR,TWORDC
*DELETE CYCLE9.235
     X  10X,32HPHASE 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  1H1)
```

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.486
        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 = ABC(1,MLT,9)
     XMAG = ABC(2,MLT,9)
     TRANS = POLD
     IF(TRANS.LE.1.0) GO TO 359
     PNEW = 0.0
     DO 356 IZ=1,NOH
 356 PNEW = PNEW + CU(IZ)*CONJG(CU(IZ))
     PNEW = PNEW*((X(2)-X(1))**2)*(NPTS/NPY)/1000.
     IF(NREG.EQ.1.OR.NREG.EQ.2) PNEW=PNEW/WNOW**2
 359 IF(TRANS.LE.1.0) PNEW = XMAG**2
     STRANS=SQRT(POLD/PNEW)
     WRITE(6,352) ABC(1,MLT,9),ABC(2,MLT,9)
*INSERT GOL.494
     IF(TRANS.GE.1) IGNAL=1
```

6. *ID TBLUM

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

```
IDENT  TBLUM

  *IDENT TBLUM
   TBLOOM
      *DELETE TBLOOM.42
            IF(AXIAL.GT.0.0) WRITE(6,596) AXIAL
      *DELETE TBLOOM.66
           X VT = (980.665*ALFA*2.*HT/(RHO*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  RMVSPH

 *IDENT RMVSPH
  QUAL
     *INSERT QUAL.51
           IF(IWRITE.EQ.0.) GO TO 60
           NDH = NPTS*NPY
           IW = IWRITE
```

```
          IF(IWRITE.LT.0) IW = -IWRITE
          WRITE(IW) (CU(IX).IX=1.NOR).X.DRX.DRY.NIT.SAVE
          REWIND IW
          WRITE(6.59) IW.IPHASE
     59   FORMAT(//.5X.*CU HAS BEEN WRITTEN ON UNIT*.I3.* FROM QUAL*.
        X * WITH IPHASE =*.I2//)
          IF(IWRITE.GE.0) GO TO 60
          IPRNT = 1
          IF(KPLOT.GT.0) WRITE(6.3000) NETITL
          IF(KPLOT.GT.0) CALL IPLOT(KPLOT)
          IF(ISAVE.EQ.1) READ(7) (CU(IX).IX=1.NOR).X.DRX.DRY
          REWIND 7
          RETURN
     60   CONTINUE
MAIN
   *DELETE MAIN.227
    200   KPLOT=1000
          READ(5.QLOT)
```

8. *ID CHGNPT

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

a. Reoverlap the beam, letting the code find the original DCALC by setting DIBEAM = 0

b. 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

  *IDENT CHGNPT
   GOL
     *DELETE GOL.242
       150 IWEA=IWEA + 1
           IF(.NOT.INIT) GO TO 153
           READ(IN.REGRID)
           ABC(10.IWEA.6) = WORD
       153 WORD = ABC(10.IWEA.6)
     *INSERT CYCLES.6
           NPTS = NEWNPT
           NPY = NEWNPY
           NOR = NPY*NPTS
     *INSERT CYCLE9.3
         X .NEWNPT.NEWNPY.IMIRSM
     C     NEWNPT AND NEWNPY ARE THE DESIRED NUMBER OF POINTS
     C        SAMPLING THE FIELD.
     C     IMIRSM = IMIR FOR THE CVM. DEFAULTS TO IMIR = 1.
     C        IT IS USED TO RENORMALIZE THE BARE RESONATOR FEEDBACK FIELD.
     *INSERT GOL.657
           NEWNPT = 0
           NEWNPY = 0
           IMIRSM = 1
```

```
*INSERT GDL.663
      ABC(6,1,1) = NEWNPT
      ABC(7,1,1) = NEWNPY
      ABC(8,1,1) = IMIRSM
      IF(OIBEAM.EQ.0.) ABC(1,1,1) = (X(2)-X(1))*FLOAT(NPTS)
*DELETE GDL.667
      NEWNPT = ABC(6,1,1)+.01
      NEWNPY = ABC(7,1,1)+.01
      IMIRSM = ABC(8,1,1)+.01
      IF(NEWNPT.EQ.0) NEWNPT=NPTS
      IF(NEWNPY.EQ.0) NEWNPY=NPY
      XDEL = DCIRM/NEWNPT*2.
*DELETE GDL.683
      DO 62 IGN=2,NEWNPT
*DELETE GDL.684
      DO 63 MY=1,NEWNPY
*DELETE GDL.686
      DO 63 MX=1,NEWNPT
*DELETE APR26.17
      NEWNOR = NEWNPT*NEWNPY
      WNOWSQ = 1.
      IF(NREG.EQ.1.OR.NREG.EQ.2) WNOWSQ=WNOW**2
      POWA = POWA/WNOWSQ
      POWB = POWB/WNOWSQ
      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 RGAIN 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

  *IDENT MISCEX

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

 GDL
    *DELETE GDL.384,GDL.385
         1NO. OF STRUTS=,I2,2X,12H X-Y CENTER=,G10.4,1H,,G10.4,/2X,
         2 13HHUB DIAMETER=,G10.4,3X,7HTHETAS=,6G10.4)
    *DELETE GDL.884,GDL.885
      600 FORMAT(//1X,119(1H*)//33H             ITERATION IS CONVERGED ,
         X   6H AFTER,I4,16H ITERATIONS    //1X,119(1H*)//)
          READ(9) (CU(IZ),IZ=1,NOR),X,DRX,DRY
          REWIND 9
 LISTRO
    *DELETE CORR1.23,CORR1.24
         =1X,4HCARD,15X,10(1H1),10(1H2),10(1H3),10(1H4),10(1H5),
         =10(1H6),10(1H7),1HH,/7H COLUMN,4X,8(10H1234567890),5X,
 GDI
    *DELETE JITTER.83,JITTER.86
          ARC(1,IJTR,5) = JITANG*1.E-6
          ARC(2,IJTR,6) = JITDIS
      231 SIGXY = ARC(1,IJTR,6)*ARC(2,IJTR,6)
          WRITE (6,1836) ARC(1,IJTR,6),SIGXY
 CAVITY
    *DELETE RAWF.11
      100 WRITE(7) (CU(IZ),IZ=1,NOR)
```

```
FILT
    *DELETE CYCLE9.119,CYCLE9.120
       25 KOUNT = KOUNT+1
          IF(IPS.EQ.2) GO TO 54
    *INSERT CYCLE9.99
          FOCAL = 1.E50
STEP
    *INSERT STEP.40
          RADCUR = RADCAR
          IF(DELZ.EQ.0.) RETURN
GOL
    *DELETE GOL.827
    *DELETE HARE.86
          IF(RGAIN) CALL REGAIN(NCT,NITER)
MAIN
    *INSERT MAIN.13
          LOGICAL RGAIN
    *INSERT HARE.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 HARE.1

         & ,IBARE,PLOTS,RGAIN,IRSDS
    *INSERT MAIN.7
          COMMON /SVTYM/ RGAIN,IFLGAP
GOL
    *INSERT GOL.17
          COMMON /SVTYM/ RGAIN,IFLGAP
    *DELETE GOL.861
          IF(ICNTL.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 (Rλ/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 Rλ/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 1.0Rλ/D quality.

IDENT EXQUAL

```
 *IDENT EXQUAL
 QUAL
    *INSERT JITTER.124
    C *** UMAX IS THE FAR FIELD CENTERLINE INTENSITY DUE TO A PLANE WAVE
    C      APERTURED TO A DIAMETER DR 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 /STROWL/ IPRNT
 GOL
    *INSERT GOL.120
         X .IPRNT
    *INSERT GOL.138
    C
    C      IPRNT IS A FLAG FOR PRINTING NREG AND WNOW FROM STEP
    C            = 0  DON'T PRINT
    C            = 1  PRINT (DEFAULT)
    *INSERT GOL.581
          IPRNT = 1
    *INSERT GOL.590
          ARC(8,ISTEP,3) = IPRNT
    *INSERT GOL.593
          IIPRNT = ARC(8,ISTEP,3) + .001
    *INSERT GOL.18
          COMMON /STROWL/ IIPRNT
 QUAL
    *INSERT JITTER.126
          IF(KPLOT.GT.0) WRITE(6,3000)  FFTITL
     3000 FORMAT(1H1,20A4//)
          IF(KPLOT.NE.0) CALL IPLOT(KPLOT)
          IPRNT = 1
          IF(IRYTFF.NE.0) WRITE(IRYTFF) (CU(IX),IX=1,NOR),X,DRX,DRY,NIT,SAVE
          IF(IRYTFF.NE.0) REWIND IRYTFF
          IF(IRYTFF.NE.0) WRITE(6,800) IRYTFF
      800 FORMAT(10X,*FAR FIELD HAS BEEN WRITTEN TO UNIT*,I4)
 STEP
    *INSERT STEP.6
          COMMON /STROWL/ IPRNT
 CAVITY
    *INSERT CAVITY.9
          COMMON /STROWL/ IPRNT
    *INSERT CAVITY.102
          IPRNT = 1
 QUAL
    *INSERT JITTER.104
          DATA FFTITL /14*4H     ,4HFAR ,4HFIEL,4HD PL,4HOTS ,2*4H    /
          DATA NFTITL /14*4H     ,4HOPTI,4HMIZE,4HD FI,4HELD ,2*4H    /
          DATA SAVE /10*0./
          IPRNT = 0
          RR1 = 1.0
          RR5 = 5.0
    *INSERT QUAL.14
    C *** ISAVE = 9 : READ IN FAR FIELD FROM UNIT 9.
    *INSERT QUAL.18
    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,RB1,PRR)
      IF(ISTEP.EQ.6) CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RB5,PRR5)
      IF(ISTEP.EQ.6.AND.RB.NE.1.)
     X CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RB,PRRRB)
*DELETE QUAL.121,QUAL.127
      WRITE(6,132) RB1,PRK,XCINT,YCINT,RB1,PRK,UMXK,XPEAK,YPEAK,
     X                  PWSAVK,DB,STREHL
      IF(RB.NE.1.)
     X WRITE(6,132) RB,PRRRBK,XCINT,YCINT,RB,PRRRBK,UMXK,XPEAK,YPEAK,
     X                  PWSAVK,DB,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=,G12.4,6X,22H   REFERENCE DIAMETER=,F6.2,/
     X 19H STREHL INTENSITY =,G11.4/)
      HQ5INT = PRR5K/PWSAVK*100.
      HQ5CNT = PRR5K/PWSAVK*100.
      WRITE(6,6010) HQ5INT,HQ5CNT
 6010 FORMAT(/10X,*NOTE: CENTROID AND IMAX COORDINATES ARE IN*,
     X * CENTIMETERS*//*  HQ ABOUT IMAX FOR 5RL/D=*,G12.4,
     X 10X,*  HQ ABOUT THE CENTROID FORM 5RL/D=*,G12.4/)
*DELETE QUAL.118
      CALL POWWOW(NPTS,DX,X,US,XCINT,YCINT,RB1,PRR)
      ZLDSQ = ZLD*ZLD
*INSERT QUAL.120
      IF(ISTEP.NE.6) GO TO 2000
C *** FIND POWER IN 5RL/D.
      CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RB5,PRR5)
      CALL POWWOW(NPTS,DX,X,US,XCINT,YCINT,RB5,PRB5)
      PRR5  = PRR5*ZLDSQ
      PRR5K = PRR5/1000.
      PRB5  = PRB5*ZLDSQ
      PRB5K = PRB5/1000.
      IF(RB.EQ.1.) GO TO 2000
C *** FIND POWER IN RB*RL/D
      CALL POWWOW(NPTS,DX,X,US,XPEAK,YPEAK,RB,PRRRB)
      CALL POWWOW(NPTS,DX,X,US,XCINT,YCINT,RB,PRBRB)
      PRRRB  = PRRRB*ZLDSQ
      PRRRBK = PRRRB/1000.
      PRBRB  = PRBRB*ZLDSQ
      PRBRBK = PRBRB/1000.
C *** RETURN TO CENTIMETERS FOR OUTPUT.
 2000 XCINT = XCINT*ZLD
      YCINT = YCINT*ZLD
      XPEAK = XPEAK*ZLD
      YPEAK = YPEAK*ZLD
      PH(ISTEP) = PRK
      HQINT = PRR/PWSAVE*100.
      HQCNT = PRB/PWSAVE*100.
      IF(PROP.NE.0.0) GO TO 360

*DELETE CYCLE9.44,CYCLE9.49
      IF(ISTEP.EQ.1) WRITE(6,5910)
 5910 FORMAT(//33X,30H   FLUX(HQ) IN 1*RL/D ABOUT    ,/
     X 12H TRIAL FOCAL,8X,5HTOTAL,6X,30H         IMAX         CENT      ,
     X 8X,9H STREHL   ,/
     X 12H   LENGTHS  ,8X,5H FLUX,6X,30H (XPEAK,YPEAK)   (XCINT,YCINT) ,
     X 9X,9HINTENSITY/1X,79(1H=))
      WRITE(6,5920) ISTEP,F,PWSAVK,PRK,HQINT,PBK,HQCNT,STREHL,
     X                  XPEAK,YPEAK,XCINT,YCINT
 5920 FORMAT(/3H  F,I1,1H=,G12.4,2X,F7.2,7X,1H ,F7.2,1H(,F4.1,1H),2X,
     X F7.2,1H(,F4.1,2H) ,6X,F9.4/33X,1H(,F6.3,1H,,F6.3,1H),
```

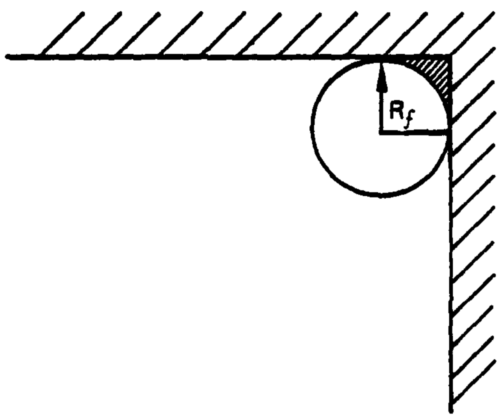
```
     X 1X,1H(,F6.3,1H,,F6.3,1H))
*DELETE CYCLE9.17
     X ,FHM(5),P(6),PB(6),XSAVE(128),FFTITL(20),NFTITL(20),SAVE(10)
*DELETE CYCLE9.61,CYCLE9.65
      POPTB = -100.
C *** FIND LOCATION OF MAXIMUM QUALITY ABOUT IMAX.
      DO 370 I=1,5
      IF(P(I).LE.POPT) GO TO 370
      POPT = P(I)
      ISV = I
  370 CONTINUE
C *** FIND LOCATION OF MAXIMUM QUALITY ABOUT THE CENTROID.
      DO 375 I=1,5
      IF(PB(I).LE.POPTB) GO TO 375
      POPTB = PB(I)
      ISVB = 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=ISVB
      IF(ICTRD.NE.3) GO TO 380
      IOPT = ISV
      IF(POPTB.GT.POPT) IOPT = ISVB
  380 FOPT=FBM(IOPT)
*DELETE QUAL.131
      IF(ICTRD.EQ.1.OR.((ICTRD.EQ.3.OR.ICTRD.EQ.0).AND.PRB.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(/2X,* THE FOLLOWING QUALITY TABLE IS FOUND ABOUT*,
     X * COORDINATES (*,G12.4,1H,,G12.4,*) FOR F =*,G12.4)
      XCINT = XCINT/ZLO
      YCINT = YCINT/ZLO
      CALL PLTOT(NPTS,DX,X,UMAX,4,,US,IPLT,
*DELETE JITTER.103
      SUBROUTINE QUAL (IPHASE,ISAVE,IPLT,TITLE,RH,ANS,DR,RF,SIGANG,PROP
     X                       KPLOT,IWRITE,ITABLE,ICTRD,IRYTEE,NIT)
MAIN
*DELETE JITTER.2
      NAMELIST/ULOT/TITLE,IQLT,DR,ISAV,IPHASE,RRR,RF,SIGANG,PROP,
     X KPLOT,IWRITE,ITABLE,ICTRD,IRYTEE
*INSERT 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 IW = IWRITE
C     IWRITE.LT.0 SETS IW =-IWRITE AND RETURNS TO 500 IMMEDIATELY.
C     IRYTEE.GT.0 WRITES THE FAR FIELD TO UNIT IRYTEE
C     KPLOT.GT.0  PLOTS THE FAR FIELD BY CALLING IPLOT(KPLOT)
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                IRL/D 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 QUALTIY TABLE ABOUT IMAX FOR OPTIMUM.
C            = 3 FIND THE OPTIMUM VALUE ABOUT EITHER CENTROID OR IMAX
C                CHOSEN FOR THE HIGHEST IRL/D QUALITY.
```

```
*DELETE JITTER.3
  210 CALL QUAL (IPHASE,ISAV,IQLT,TITLE,RHH,AS,DH,RF,SIGANG,PROP,
     X KPLOT,IWRITE,ITABLE,ICTRD,IRYTEE,NIT)
      PROP = 0.0
      KPLOT = 0
      RH =1.
      IWRITE = 0
      IRYTEE = 0
      ITABLE = 0
      ICTRD  = 0
      IFLGAP = 1
*INSERT MAIN.22
      DATA SIGANG /0.0/
      DATA PROP /0.0/
      DATA KPLOT,IWRITE,IRYTEE /0,0,0/
      DATA ITABLE,ICTRD /0,0/
PLTOT
*INSERT PLTOT.53
      WRITE(50,2000) TITLE
 2000 FORMAT(1X,20A4,//,4X,*RL/D*,5X,*FRACTION*/)
*INSERT APR26.33
      DO 2025 I=1,30
 2025 WRITE(50,2024) RRD(I),PWA(I)
 2024 FORMAT(3X,F4.1,5X,F8.5)
```

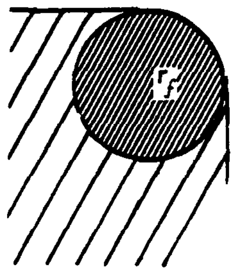
11. *ID FILAPR

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 RFMIN 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

     ID 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.EQ.1) GO TO 50
     C ***** IIN = 0  FOR  OUTER APERTURE
            AMXOUT = AMAX1(AINT,AMXOUT)
            IF(AINT.NE.AMXOUT) GO TO 60
            XMXOUT = X
            YMXOUT = Y
            GO TO 60
```

```
C ***** IIN = 1   FOR  INNER APERTURE
   50  AMAXIN = AMAX1(AINT.AMXIN)
       IF(AINT.NE.AMXIN) GO TO 60
       XMXIN = X
       YMXIN = Y
   60  CONTINUE
*DELETE APRFIX.88.APRFIX.91
       IF(IIN.EQ.1) GO TO 70
C ***** IIN = 0   FOR  OUTER APERTURE
       AMAXOUT = AMAX1(AINT.AMXOUT)
       IF(AINT.NE.AMXOUT) GO TO 80
       XMXOUT = X
       YMXOUT = Y
       GO TO 80
C ***** IIN = 1   FOR  INNER APERTURE
   70  AMXIN = AMAX1(AINT.AMXIN)
       IF(AINT.NE.AMXIN) GO TO 80
       XMXIN = X
       YMXIN = Y
   80  CONTINUE
*DELETE APRFIX.102.APRFIX.105
       AMXIN = AMXIN*EXF/1000.
       AMXOUT = AMXOUT*EXF/1000.
       IF(RDISK.NE.0..OR.YDISK.NE.0.) WRITE(6.310) AMXIN.XMXIN.YMXIN
  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(RAPRTR.NE.0..OR.YAPRTR.NE.0.) WRITE(6.320)AMXOUT.XMXOUT.YMXOUT
  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 APRFIX.1
       SUBROUTINE APRTR(RPRTRI.RDISKI.XPOS.YPOS.YAPRTR.YDISK.
     X XAPRTR.XDISK)
*DELETE APRFIX.8.APRFIX.8

C      MODIFIED 3/8/77 BY R. EILEGER FOR RECTANGULAR APERTURE OF
C      WIDTH=2*XAPRTR AND HEIGHT=2*YAPRTR AND A CENTRAL
C      OBSCURATION RATIO OF WIDTH=2*XDISK AND HEIGHT=2*YDISK.
C      WHEN RECTANGULAR APERTURES (OR SQUARE) ARE USED. RAPRTR AND
C      RDISK BECOME RADII OF CURVATURE FOR FILLETING THE APERTURE
C      AND CENTRAL OBSCURATION CORNERS RESPECTIVELY.
*INSERT APRFIX.15
       RAPRTR = RPRTRI
       RDISK = RDISKI
*DELETE SQAPR.3.SQAPR.6
       HO = 2.*YAPRTR
       HI = 2.*YDISK
       WO = 2.*XAPRTR
       WI = 2.*XDISK
*DELETE SQAPR.8.SQAPR.13
 1000  FORMAT(/28H   CIRCULAR APERTURE APPLIED //
     A         19H    OUTSIDE RADIUS = . G12.4/
     B         19H    INSIDE  RADIUS = . G12.4)
 1001  FORMAT(/31H   RECTANGULAR APERTURE APPLIED //
     A   26H   OUTSIDE DIMENSIONS ARE .G12.4.9H HIGH BY .G12.4.5H WIDE/
     B   26H   INSIDE  DIMENSIONS ARE .G12.4.9H HIGH BY .G12.4.5H WIDE)
       IF(RAPRTR.NE.0.) WRITE(6.1004) RAPRTR
 1004  FORMAT(26H   FILLET RADIUS          = .G12.4)
       IF(RDISK.NE.0.) WRITE(6.1005) RDISK
 1005  FORMAT(26H    OBSCURATION RADIUS    = .G12.4)
       WRITE(6.1003) XPOS.YPOS
 1003  FORMAT(10H   XPOS = .G12.4./10H   YPOS = .G12.4//)
       IF (YAPRTR.EQ.0.0) GO TO 260
```

```
*INSERT SQAPR.1
      WRITE(6,1003) XPOS,YPOS
*DELETE APREIX.32
      IF(IIN.EQ.0.AND.R.GE.RAPRTR) INTCK=1
      IF(IIN.EQ.1.AND.R.LE.RDISK) INTCK=1
*DELETE APREIX.63,APREIX.64
      A = XAPRTR
      B = YAPRTR
      AS = A - RAPRTR
      BS = B - RAPRTR
      RAD = RAPRTR
*DELETE APREIX.70
      IF((ABS(X).GE.XAPRTR.OR.ABS(Y).GE.YAPRTR).AND.IIN.EQ.0) INTCK = 1
      IF((ABS(X).LE.XDISK.AND.ABS(Y).LE.YDISK).AND.IIN.EQ.1) INTCK = 1
*INSERT APREIX.77
      IF(XMIN.GE.AS.AND.YMIN.GE.BS) GO TO 400
*DELETE APREIX.79
      IF(XMAX.LE.A.OR.YMAX.LE.B) GO TO 200
*DELETE APREIX.90
      IF(XMAX.LE.A.OR.YMAX.LE.B) GO TO 200
*DELETE APREIX.94,APREIX.98
  260 IF(YDISK.EQ.0..OR.IIN.EQ.1) GO TO 300
      IIN = 1
      A = XDISK
      B = YDISK
      AS = A - RDISK
      BS = B - RDISK
      RAD = RDISK
      GO TO 149
  400 CONTINUE
      XF = ABS(X) - AS
      YF = ABS(Y) - BS
      XF = SIGN(XF,X)
      YF = SIGN(YF,Y)
      R = SQRT(XF**2 + YF**2)
      IF(R.GE.RAPRTR.AND.IIN.EQ.0) INTCK=1
      IF(IIN.EQ.1.AND.R.LE.RDISK) INTCK=1
      RPP = RO(XF,YF, 1, 1)
      RMM = RO(XF,YF,-1,-1)
      RMP = RO(XF,YF,-1, 1)
      RPM = RO(XF,YF, 1,-1)
      PER = 1.
      RMAX = AMAX1(RPP,RMM,RMP,RPM)
      IF(RMAX.LE.RAD) GO TO 200
      PER = 0.
      RMIN = AMIN1(RPP,RMM,RMP,RPM)
      IF(RMIN.GE.RAD) GO TO 200
      PER = (RAD-RMIN)/(RMAX-RMIN)
      GO TO 200
GOL
*INSERT ROT.4
     X ,REMIN,REMOUT
*DELETE SQAPR.22
      NAMELIST /APTURZ/ DOUT,DIN,XPOS,YPOS,YOUT,YIN,REIN,REOUT
*INSERT ROT.3
      DATA REIN,REOUT,REMIN,REMOUT /4*0./
*INSERT GOL.114
C     REMIN  = RADIUS OF CENTRAL OBSCURATION CORNER.
C     REMOUT = RADIUS OF FILLET.
*INSERT GOL.145
C     REIN  = RADIUS OF CENTRAL OBSCURATION.
C     REOUT = RADIUS OF FILLET.
```

```
*INSERT RAHE,33
      REMIN  = 0.
      REMOUT = 0.
      RTST   = 0.
*INSERT ROT,8
      AHC(12,IMIR,4) = 0.
      AHC(13,IMIR,4) = 0.
      IF(DOUTY.NE.0..OR.DINY.NE.0.) RTST = 1.
      IF(RTST.EQ.0) GO TO 22
      AHC(4,IMIR,2) = REMOUT
      AHC(5,IMIR,2) = REMIN
      AHC(12,IMIR,4) = DIAOUT/2.
      AHC(13,IMIR,4) = DIAIN/2.
*DELETE RAHE,60,RAHE,61
      IM = AHC(8,1,1) + .01
      CALL APRTR  (ABC(4,IM,2), AHC(5,IM,2), AHC(6,IM,2), ABC(7,IM,2),
     X AHC(10,IM,4), AHC(11,IM,4), ABC(12,IM,4), AHC(13,IM,4))
C   *** THE ABOVE ASSUMES THAT CVM IS MIRROR NUMBER IM.
*DELETE GOL,543,RAHE,62
  21  CALL MIRROR    (AHC(1,IMIR,2), AHC(2,IMIR,2), AHC(3,IMIR,2),

     X AHC(4,IMIR,2), AHC(5,IMIR,2), AHC(6,IMIR,2), AHC(7,IMIR,2),
     X AHC(8,IMIR,2), ABC(9,IMIR,2), AHC(10,IMIR,2),AHC(11,IMIR,2),
     X ABC(12,IMIR,2),AHC(13,IMIR,2),AHC(14,IMIR,2),
     X AHC(10,IMIR,4),ABC(11,IMIR,4),AHC(12,IMIR,4),ABC(13,IMIR,4))
MIRROR
*DELETE SOAPR,38
      CALL APRTR(RIAOUT,RIAIN,XPOS,YPOS,RYOUT,RYIN,RXOUT,RXIN)
*DELETE MIRROR,2,ROT,10
      SUBROUTINE MIRROR(ANX,ANY,RADC,RIAOUT,RIAIN,XPOS,YPOS,RFL,DELTM,
     X DISTF,RANGLS,PHIAST,PHIRT,DESIP,
     X RYOUT,RYIN,RXOUT,RXIN)
C     . THE FIRST 2 LINES ARE ABC(N,IMIR,2)   N=1,14  AND
C        AND THE LAST LINE IS  AHC(N,IMIR,4)   N=10,13
*DELETE MIRROR,13
      IF(RIAOUT.EQ.0..AND.RIAIN.EQ.0..AND.RYOUT.EQ.0..AND.RYIN.EQ.0.)
     X GO TO 70
GOL
*DELETE GOL,546
      ROUTY = AHC(10,IMIR,4)
      ROUTX = AHC(4,IMIR,2)
      IF(ROUTY.NE.0) ROUTX = AHC(12,IMIR,4)
      RAPTR = ROUTX
      IF(ROUTY.NE.0) RAPTR=AMIN1(ROUTX,ROUTY)
*INSERT GOL,616
      REIN  = 0.
      REOUT = 0.
      RTST  = 0.
*INSERT SOAPR,34
      AHC(7,IAP,4) = 0.
      AHC(8,IAP,4) = 0.
      IF(YOUT.NE.0..OR.YIN.NE.0.) RTST = 1.
      IF(RTST.EQ.0) GO TO 41
      ABC(1,IAP,4) = REOUT
      AHC(2,IAP,4) = REIN
      AHC(7,IAP,4) = DOUT/2.
      ABC(8,IAP,4) = DIN/2.
*DELETE SOAPR,35,GOL,626
      IF(DOUT.GE.0.0.AND.DIN.GE.0.0)
     X CALL APRTR(ABC(1,IAP,4),ABC(2,IAP,4),AHC(3,IAP,4),AHC(4,IAP,4),
     X             AHC(5,IAP,4),AHC(6,IAP,4),AHC(7,IAP,4),ABC(8,IAP,4))
      RASAVE = RAPTR
```

```
      IF(YOUT.NE.0.) RAPTR = ATAN2(DOUT.YOUT)/2.
      IF(YOUT.EQ.0.) RAPTR = DOUT/2.
      IF(RAPTR.LE.0.0) RAPTR = RASAVE
*DELETE GDL.622
   41 PPW=0.
      DO 13 IZZ = 1.NOB
   13 PPW = PPW + CU(IZZ)*CONJG(CU(IZZ))
      SPPW = PPW*(X(2)-X(1))**2*(NPTS/NPY)
      IF(NREG.EQ.1.OR.NREG.EQ.2) SPPW=SPPW/WNOW**2
      DOUT = 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.) DOUT= AHC(7.IAP.4)*2.
      IF(YOUT.NE.0..OR.YIN.NE.0.)  DIN= AHC(8.IAP.4)*2.
      IF(DOUT.LT.0.0.AND.DIN.LT.0.0)
```

12. *ID NUDISKT

Ident NUDISKT 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   NODISKI

  *IDENT NODISKI
   GOL
      *DELETE GOL.75
      C              = 16   GO PUNCHED ON CARDS. READS PUNCH.
      *INSERT GOL.49
             DATA  KWRITE.KREAD.READ3 /0.0..F./
      *INSERT GOL.184
             NAMELIST /PUNCH/ KREAD.KWRITE
      C          THIS IS A FORMATTED VERSION OF DISKIT.
      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.395
        160 KREAD = 0
            KWRITE = 0
            READ(IN.PUNCH)
            IF(KREAD.EQ.0.AND.KWRITE.EQ.0) GO TO 999
            IF(KREAD.EQ.0) GO TO 169
            READ(KREAD.165) TITLE
        165 FORMAT(20A4)
            WRITE(6.166) KREAD.TITLE
        166 FORMAT(2X.*FORMATTED 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.DUME1.X(I+1).X(J).DUM2.DUME2
            II = 2*(I+IREF)
            CUR(II-1) = DUM1
            CUR(II) = DUME1
            CUR(II+1) = DUM2
            CUR(II+2) = DUME2
        167 CONTINUE
        168 FORMAT(2F8.3.2F12.6.2F8.3.2F12.6)
            REWIND KREAD
            GO TO 999
        169 WRITE(6.163) KWRITE
        163 FORMAT(2X.*FORMATTED FIELD WRITTEN TO UNIT :*.I3.*.*)
            WRITE(KWRITE.164) (GNOT(ICNTL.I).I=1.20)
     *DELETE GOL.403.GOL.406
        161 WRITE(KWRITE.168) X(I).X(J).DUM1.DUME1.X(I+1).X(J).DUM2.DUME2
            REWIND KWRITE
     *DELETE GOL.178
             NAMELIST /DISKIT/ IREAD.IWRITE.IORD.IADD.READ3.RWIND
     C           READ3 = .T. MEANS READ OR WRITE TO TAPE IN THREE STEPS.
     C           RWIND = .T. MEANS REWIND WRITTEN(READ) TAPE.
     *INSERT GOL.455
             READ3S(NOS) = READ3
             RWINDS(NOS) = RWIND
             IF(READ3S(NOS)) GO TO 104
     *INSERT GOL.460
             IF(READ3S(NOS)) GO TO 104
     *INSERT GOL.459
             READ3 = .F.
             RWIND = .T.
     *DELETE GOL.462

     IR = IREAD
     IF(IREAD.LT.0) IR = -IREAD
     IF(IREAD.GT.0) READ (IR) (CU(IZ).IZ=1.NOR).X.DRX.DRY.NITER
     IF(IREAD.LT.0) READ (IR) (CU(IZ).IZ=1.NOR).X.DRX.DRY.NITER.SAVE
```

```
*DELETE GOL.472
      IR = IREAD
      IF(IREAD.LT.0) IR = -IREAD
      IF(IREAD.GT.0) READ (IR) (CU(IZ),IZ=1,NOR),X,DRX,DRY,NITER
      IF(IREAD.LT.0) READ (IR) (CU(IZ),IZ=1,NOR),X,DRX,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(5,104)
  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 DTITL(NDS,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(J),J=1,NPXIN)
      DO 123 J=1,NPXIN
      JM1 = (J-1)*NPXIN
  123 READ(IREAD) (CU(I+JM1),I=1,NPXIN)
      NPTS = NPXIN
      NPY  = NPYIN
      IF(RWIND) REWIND IREAD
      IF(.NOT.RWIND) WRITE(6,104)
      GO TO 999
  110 READ(IN,1243) TITLE
      DO 125 L=1,20
  125 DTITL(NDS,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(J),J=1,NPTS)
      IF(NPTS.EQ.NPY) GO TO 140
C *** UNFOLD CU
      DO 130 J=1,NPY
      JM1 = (J-1)*NPTS
      JIM1 = (NPTS-J)*NPTS
      DO 130 I=1,NPTS
      IJ = I+JM1
      IJI = I+JIM1
  130 CU(IJI) = CU(IJ)


  140 DO 141 J=1,NPTS
      JM1 = (J-1)*NPTS
  141 WRITE(IWRITE) (CU(I+JM1),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 namelist CAVTY2.

(2) Account for the gain length by the factor ZFACT, also in CAVTY2.

```
IDENT   CY4KIN

  *IDENT CY4KIN

       ID CY4KIN INCORPORATES THE CYCLE IV KINETICS PACKAGE FROM
          PWA/GPD IN WEST PALM BEACH, FLORIDA.  OXYGEN AND HYDROGEN
          KINETICS WERE ADDED AS WELL AS THE ABILITY TO SIMULATE
          R-BRANCH TRANSITIONS ON THE 9.4 MICRON BAND.

  BLOMIT
     *DELETE BLOMIT.11
          5    TITLE(20),VG(5),TVO2(5), FH2(5), NSYM
  CAVITY
     *DELETE CAVITY.16
          6                VG(5),TVO2(5),FH2(5),NSYM
     *INSERT LROP1.10
           DATA XH2 /0./
     *DELETE LROP1.11
          1 AVGAIN,GFACT2,TO2,XH2,GDC,ZFACT
     *DELETE CAVITY.75,CAVITY.77
     C       T1   IS VIBRATIONAL TEMPERATURE OF V00    AT NEP, DEG K
     C       T2   IS VIBRATIONAL TEMPERATURE OF 0V0    AT NEP, DEG K
     C       T3   IS VIBRATIONAL TEMPERATURE OF 00V    AT NEP, DEG K
     *DELETE CAVITY.92
     C       BRNCH IS THE J VALUE OF THE LOWER LASER LEVEL FOR THE TRANSITION
```

```
*INSERT CAVITY.47
C       XH2  IS MOLE FRACTION OF HYDROGEN
*DELETE CAVITY.127
        WRITE (6,103) XN2,XCO2,XH2O,XCO,XO2,XH2,GDC
*DELETE CAVITY.129
      X 5HXH2O =,G12.5,4X,5HXCO =,G12.5,4X,5HXO2 =,G12.5,4X,5HXH2 =,
      X G12.5,/* GLADSTONE-DALE CONSTANT     = *,G12.5)
*INSERT CAVITY.134
        TVO2(NCAVN)=TO2
*INSERT CAVITY.138
        WRITE(6,1104) TO2
 1104 FORMAT(1X,6HTO2 = ,G12.5)
*INSERT CAVITY.174
        IF(XO2.NE.0..AND.TO2.EQ.0.)  STOP "NEED TO2"
        FH2(NCAVN) = XH2
DENSY
*INSERT DENSY.30
        COMMON /GLAD/ STONE,ZFACTR
*INSERT DENSY.100
        GDC = STONE
CAVITY
*INSERT LROPI.9
        COMMON /GLAD/ STONE,ZFACTR
GAINXY
*INSERT GAINXY.15
        COMMON /GLAD/ STONE,ZFACTR
*INSERT GAINXY.116
        DELTAZ = DELTAZ*ZFACTR
        IF(ZFACTR.NE.1.) WRITE(6,1000) ZFACTR,DELTAZ
 1000 FORMAT(/5X,5H*****,* WARNING - GAIN MODIFIED BY*,F7.4,
      X * ACTIVE LENGTH, DELTAZ =*,G12.4,5H*****)
CAVITY

*INSERT CAVITY.103
        GDC = STONE
        ZFACT = ZFACTR
*INSERT CAVITY.109
        STONE  = GDC
        ZFACTR = ZFACT
DENSY
*DELETE DENSY.31
FUHS
*DELETE FUHS.11
      6               AVG(5),TVO2(5),FH2(5),NSYM
        COMMON /GLAD/ STONE,ZFACTR
*DELETE FUHS.119
        ROCL=-STONE*RHO*ZC(NCV)/NS(NCV)
GAINXY
*DELETE GAINXY.10
        COMMON/START/ TST,PST,VT,E00VT,E0V0I,EV00I,EN2I,GAINI,EGY02I
*DELETE GAINXY.12
        COMMON /MOLES/ XN2,XCO2,XH2O,XCO,XO2,XH2
*DELETE GAINXY.14
        COMMON/RATE/ R12,RC1,RC2,RPUMP,RSTIM,R8,R9,R10
*DELETE GAINXY.20
      5    TITLE(20),AVG(5),TVO2(5),FH2(5),NSYM
*INSERT GAINXY.74
        XH2 = FH2(NCV)
*INSERT GAINXY.38
        TO2=TVO2(NCV)
*INSERT GAINXY.65
        GCON=.991E-16*RH
```

```
          EXER=-1997.
          DER=1580.
          IF(GFACT(NCV).GE.1.) GO TO 10
C    MODIFY CONSTANTS FOR R-BRANCH TRANSITION
          ROTUP=.5566*(RR+1.)*(RR+2.)
          GCON=.782E-14*(RR+1.)
          EXER=-1050.
          DER=1530.
     10 CONTINUE
*DELETE GAINXY.48
*DELETE LROP1.14,LROP1.15
*DELETE LROP1.16,LROP1.17
*DELETE LROP1.18,LROP1.19
          XMW = 28.016*XN2 + 44.011*XCO2 + 18.016*XH2O + 32.*XO2 + 2.016*XH2
          XO2FAC=18.526
*INSERT LROP1.20
        X + .34*XH2/SQRT(1.928/22.005)
*DELETE LROP1.21
*DELETE GAINXY.56
          XLAMR = 1.434/(DER+ROTUP-ROTLO)
*DELETE GAINXY.60
          GAMMA = (7.*(XN2+XCO2+XO2+XH2)+8.*XH2O)/(5.*(XN2+XCO2+XO2+XH2)+
        X   6.*XH2O)
*DELETE GAINXY.68
          CP = 3.5*R0*(XN2+XCO2+XO2+XH2+8./7.*XH2O)
*INSERT GAINXY.72

          EGYO2I = 0.0
          IF(XO2.NE.0.) EGYO2I=XO2*1556./(EXP(2239./TO2)-1.)
*INSERT GAINXY.85
          EXEROT=EXER/T1
          IF(GFACT(NCV).LT.1.) EXEROT=EXER/T2
*DELETE GAINXY.87
        X   -.561*EXP(EXEROT -ROTLO/TS))
KINET
*DELETE KINET.8,KINET.9
          COMMON/START/ TSI,PSI,VI,E00VI,E0V0I,EV00I,EN2I,GAINI,EGYO2I
          COMMON /MOLES/ XN2,XCO2,XH2O,XCO,XO2,XH2
*DELETE KINET.11
          COMMON/RATE/ RN2,RC3,RC2,RPUMP,RSTIM,R8,R9,R10
*INSERT KINET.12
          COMMON/GFACTR/GFACT(2)
*INSERT KINET.23
          EO2=XO2*1556.
          DEV0V01 = 0.
          DEN2O2 = 0.
          CHGMF = 0.
          EXER=-1997.
          IF(GFACT(1).LT.1.) F4=1.285E10/HNU
          IF(GFACT(1).LT.1.) EXER=-1950.
*INSERT KINET.36
          EGYO2=EGYO2I
*INSERT KINET.71
          IF(XO2.EQ.0.) GO TO 30
          O2EXP=EXP(-2239.044/TS)
          QEO2EQ=EO2*O2EXP/(1.-O2EXP)
          THOV0=1.-EXP(-959.8/TS)
          EROV0=(EQOV0-E0V0)/E0V0
          THN2=1.-EXP(-3354.309/TS)
          ERN2=(EQN2-EN2)/EN2
          CHGMF=(EGYO2-QEO2EQ)*R4
          THO2=1.-O2EXP
          ERO2=(QEO2EQ-EGYO2)/EGYO2
```

```
      DEO2N2=R9*EGYO2/THO2*EN2/EQN2*(THO2*ERO2-THN2*ERN2)
      DEN2O2=-3354.303/2239.*DEO2N2
      DEO2=R10*EGYO2/THO2*(EOVO/EQOVO)**2.*(1.+THO2*ERO2-(1.+THOVO*EROVO
     X   )**2.)
      DEOVO1=-2.*959.6/2239.*DEO2
      CHGO2=(-CHGAE+DEO2N2+DEO2)*DT
      EGYO2=EGYO2+CHGO2
   10 CONTINUE
*DELETE KINET.75,KINET.76
      DEGL=DEOVO-1.034*DEN2MP-1.086*DEOOVR-F4*F10*DT+DEOVO1*DT
      EN2=EN2-DEN2+DEN2O2*DT
*DELETE KINET.79
      SUMDEV=SUMDEV+(DEOVO+CHGAE*DT)*V*1.987E-16*RHON
*DELETE KINET.83
      DEV=(DEOVO/DT+CHGAE)*1.19677E8
*INSERT KINET.121
      EXFROT=EXFR/T1
      IF(GFACT(1).LT.1.) EXFROT=EXFR/T2
*DELETE KINET.123

     X  -.56)*EXP(EXFROT -ROTLO/TS))
MIX
 *DELETE MIX.6,MIX.7
      COMMON /MOLES/ XN2,XCO2,XH2O,XCO,XO2,XH2
      COMMON /RATE/ RN2,RC3,RC2,RPUMP,RSTIM,R8,R9,R10
 *DELETE MIX.28,MIX.29
 C      CO2(00V)+H2 = CO2(0V0) + H2
      TC3H = EXP(12.4*TTRO2+4.99*TTRO-2.13)
 C      CO2(0V0)+H2 = CO2 + H2
      TC2H = EXP(112.*TTRO2-57.2*TTRO+1./23)
      RC2 = PS*(XN2/TC2N+XCO2/TC2C+XH2O/TC2W+XO2/TC2O+XH2/TC2H)*1.E6
      RC3 = PS*(XN2/TC3N+XCO2/TC3C+XH2O/TC3W+XO2/TC3N+XH2/TC3H)*1.E6
 *INSERT MIX.31
      R10=XCO2/(8.E-6-TS/150.*1.E-6)   *PS
      R9=XN2/.345E-2   *PS
      R8=1./EXP(88.47*TTRO) *    ( XN2/5.4E-9 + XCO2/1.E-8 + XH2O
     X      /2.7E-13 + XO2/5.4E-9)   *PS
REGAIN
 *DELETE REGAIN.13
     5   TITLE(20),AVG(5),TVO2(5),EH2(5),NSYM
SIMPGG
 *DELETE SIMPGG.11
     5   TITLE(20),AVG(5),TVO2(5),EH2(5),NSYM
      COMMON/GFACTR/GFACT(2)
 *INSERT SIMPGG.49
      IF(GFACT(1).LT.1.) ETA = 0.45
MAIN
 *INSERT MAIN.20
      DATA GDC /0.226/,ZFACT /0.96/
 *INSERT MAIN.143
      STONE = GDC
      ZFACTR = ZFACT
 *INSERT MAIN.8
      COMMON /GLASS/ STONE,ZFACTR
```

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:

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

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 ≠ 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  MIRFIX

  *IDENT MIRFIX
   MIRROR
   *DELETE MIRROR.35
          PWRDES =1.E6
          RFLDES = .995
          RFLFAC = (1.-RFL)/(1.-RFLDES)
   *DELETE MIRROR.70
          DELTA = DELTA*PWRFAC*RFLFAC
   *INSERT MIRROR.149
          IF(NREG.EQ.1.OR.NREG.EQ.2) DELL=DELL/WNO**2
   *DELETE ROT.23
          IF(PHIAST.NE.0.) WRITE(6,420) PHIAST,RMSAG,RMTAN
   *DELETE CIOASIG.19
          IF(PHIAST.NE.0.) WRITE(6,420) PHIAST,RMSAG,RMTAN
   *DELETE CIOASIG.28
          IF(PHIAST.NE.0.) WRITE(6,420) PHIAST,RMSAG,RMTAN
   *DELETE CIOASIG.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 * 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.

$$CU_{Z''} = CU_{Z'} \cdot \frac{f}{Z} \cdot e^{-ik\,(f - Z)\;\frac{-ik}{2f}\,\vec{x}^{\,2}\left(1 - \frac{Z}{f}\right)} \qquad \text{(B3)}$$

And

$$\vec{x}'' = \frac{Z}{f}\;\vec{x}' \qquad \text{(B4)}$$

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.

```
IDENT IDPROP
 *IDENT IDPROP

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

```
QUAL
   *DELETE QUAL.41
         IF(PROP.GE.0.) GO TO 65
   C *** PROP.LT.0. MEANS APPLY A LENS OF FOCAL LENGTH PROP (CONVERGING)
   C       AND PROPAGATE TO THAT FOCAL LENGTH.
         FRATIO = -F/PROP
         F = -PROP
         ZLD = ZLD/FRATIO
         WT = WT*FRATIO*FRATIO
         DO 61 I=1,NPTS
         XSAVE(I) = XSAVE(I)/FRATIO
      61 X(I) = XSAVE(I)
         DX = DX/FRATIO
         DXSAVE = DXSAVE/FRATIO
         RKO2 = RK/2.
         DO 62 J=1,NPY
         JM1 = (J-1)*NPTS
         YSQ = X(J)**2
         DO 62 I=1,NPTS
         IIJJ = 2*(I+JM1)
         IIJJM1 = IIJJ - 1
         PHI = RKO2*(X(I)**2 + YSQ)*(FRATIO-1.)/PROP
   C  **    RECALL THAT PROP IS NEGATIVE
         COSP = COS(PHI)
         SINP = SIN(PHI)
         CURE = CUR(IIJJM1)
         CUIM = CUR(IIJJ)
         CUR(IIJJM1) = (CURE*COSP - CUIM*SINP)*FRATIO
      62 CUR(IIJJ)    = (CURE*SINP + CUIM*COSP)*FRATIO
      65 CONTINUE
   C *** CHANGE X TO DIMENSIONLESS FAR FIELD X BY DIVIDING BY ZLD=IRL/D.
   *DELETE JITTER.122
   C *** APPLY A SMALL PHASE TO CU SO THAT ITS PHASE IS NOT IDENTICALLY
   C       ZERO.
         CUR(IZIZ) = 1.E-10/FLOAT(IZIZ)
   *INSERT JITTER.119
      69 CONTINUE
   *DELETE JITTER.115
         IF(DMAX.GE.DS(IZ)) GO TO 69
 RUL
   *DELETE JITTER.97
   C *** APPLY A SMALL PHASE TO CU SO THAT ITS PHASE IS NOT IDENTICALLY
   C       ZERO.
     235 CUR(IJIJ) = 1.E-10/FLOAT(IJIJ)
 QUAL
   *INSERT CYCLE9.30
         IF(PROP.NE.0.) ISTEP=6
   *INSERT CYCLE9.36
         IF(PROP.NE.0.0) GO TO 360
   *DELETE CYCLE9.51,CYCLE9.52
5904 WRITE(6,5940) F,ZLD
5940 FORMAT(//9X,*OPTIMUM RESULTS AT F =*,G12.4,2X,* WITH  *,
    X 14H1*R*LAMBDA/D) =,G12.4,1H:)
 360 IF(PROP.NE.0.0) WRITE(6,5941) F,ZLD
5941 FORMAT(//20X,*PROPAGATION RESULTS FOR F =*,G12.4,10X,* WITH  *,
    X 14H1*R*LAMBDA/D =,G12.4,1H:)
```

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  ECSFIX

  *IDENT ECSFIX
   STEP
      *INSERT STEP.7
           X  ,APR
            COMMON /STPLC4/ APR
   DENSY
      *DELETE DENSY.23,CORR2.5
            COMMON /MELT/ R(20000), X4(21), Y4(21,81), Z4(21,81),
           X  C4(21,81), M4(21), N4, ROCL, DUMYS(40778)
   CAVITY
      *DELETE CORR1.43
      *DELETE C100ENS.3
           X POD(2),XCAV(1,0),CUR(32768),US(17100)
      *DELETE CAVITY.22
            EQUIVALENCE (CU(1),CUR(1)), (CG(1),US(1))
   REGAIN
      *DELETE SOQ77CY1.189
      *DELETE C100ENS.33
            COMMON /GLAD/ STONE,ZFACTR
            DIMENSION POD(17100), P(17100), G(17100)
      *DELETE SOQ77CY1.190,C100ENS.34
            EQUIVALENCE (POD(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
         SEGMETED LOADING OF THE DECK.

   DENSY
      *INSERT DENSY.22
           COMMON /SEGDEN/ TM8,M8,XM8,MM8,Y8,Z8,Y8W,Z8W,
          X TM9,M9,XM9,MM9W,Y9,Z9,Y9W,Z9W
```

```
CAVITY
    *INSERT CAVITY,10
          COMMON /SEGCV2/ XLEN,YLEN,ZLEN,XMCAV,YMCAV,NOUX,NODY,NOSEG,
         X FLAG,NRFST,NGTYPE,NGPLOT,IUSE,IPOEN,T1,T2,T3,IN2,TS,PS,V,
         X PHRCH,XN2,XCO2,XH2O,XCO,XO2,ALFA,ACH,VELTY,TTEMP,ANGL,
         X AVGAIN,GFACTR,TO2,XH2,PI,NOR,NNSYM,CAKAY,TOPIWL
GOL
    *INSERT GOL,16
          COMMON /SEGGOL/ IFLOW,GNOTE,IPLOTS,KPLOT,
         A NCAVNO,ILR,NSTF,NPLT,ZPROPI,ZPROPO,
         B ANGAX,ANGYY,RADC,DIAOUT,DIAIN,XMPOS,YMPOS,RMIR,REMIN,REMOUT,
         C DELTA,DISTF,DOUTY,DINY,RANULS,PHIASI,PHIROT,DESIPW,
         D DELZ, ROCURV,WINDOX,WINDOK,IIFG,IITR,IIPS,
         E DOUT,DIN,XPOS,YPOS,YOUT,YIN,REOUT,REIN,
         F DIPFAM,OVPLAP,DXXR,DYYR,MAXIT,AVCUSM,CUSMF,NEWNPT,NEWNPY,IMIRSM,
         G TITLE,RADPLT,DSM,REMOVE,PHIARR,RO,R,PERNG,
         H ALFA,SCR,T,RHO,ZLEN,NSTEPS,INPT,NPROP,AXIAL,UT,
         I IREAD,IWRITE,IOR,IADD,READ3,RWIND,
         J TRANS,XMAG,NBEAM,AWL,NGRO,JITANG,JITOIS,KREAD,KWRITE,
         K ALPHAN,CONMIR,ALPHAG,PHOGAS,TAU,TIN,REFMIR,CONGAS,
         L JSPD,WIDTH,THETA,XSPC,YSPC,DIH,CAPR,EXPAND,ROC,DISP,TILT,
         M DELZR,DELZTH
MAIN
    *DELETE CORR1,1,SOQ77CY1,2
          PROGRAM SOQ(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 SOQ CODE CAN BE RUN USING THE SEGMENTED
C     LOADER ON THE CYBER 176 COMPUTER.  THE CRITICAL JCL IS
C         ATTACH,LGO,RSOQ7912R,ID=*********.
C         ATTACH,TREE,SOQSEGTREE,ID=*********.
C         SEGLOAD,I=TREE.
C         LGO.
C      THE FILE SOQSEGTREE CONTAINS THE FOLLOWING CARDS:
C                GLOBAL      FST,MELT,CAV2,CAVX,CCG,BARES,GLAD
C                GLOBAL      PLTSIG,INITL,WAY,MRPROP,GFACTR,LENSY
C                GLOBAL      START,PROPT,MOLES,ENERG,RATE,STPOWL
C                GLOBAL      FACTER,OAZ,TIME,VIEW,CG,STPLCM,SVTYM
C                GLOBAL      FCL,C,,OR,IO,,AOR,RM,CON,RM,PUT,FO,TERM,RM
C                GLOBAL      SKFL.FO,SKSF.FO,JMPS.RM,OPFS.FO
C   CAVITY       GLOBAL      SEGCV2-SAVE

C   GOL          GLOBAL      SEGGOL-SAVE
C   GOL          GLOBAL      ZIP-SAVE
C                TREE        SOQ-(QUAL,SGOL,LISTRO)
C   SGOL         TREE        GOL-(CAVITY,FIFLOS,MIRROR,REGAIN,RSTEP,SLIVER,
C   ,THLOOM,AXICN,ZERN)
C                INCLUDE      CHEK.RM,PTWR.SQ,SKSF.SQ,OPES.SQ
C                INCLUDE     STEP,FOURT,TILT,RACKSP=,EOF,ATAN,SPTAN
C                INCLUDE     ACOSIN=,COS=SIN,SINCOS=
C   GOL          INCLUDE     INTERP
C   SLIVER       INCLUDE     SPIDER
C   THLOOM       INCLUDE     THERML
C                END
C
C        WHERE THE LEFT HAND COLUMN STARTS IN COLUMN 1.
C
```

18. *WNDOW

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   WNDOW

  *IDENT WNDOW
   GDL
     *DELETE GDL.481
           CALL AEROW
 AEROW
     *DELETE AEROW.2
           SUBROUTINE AEROW
     *DELETE AEROW.6.AEROW.7
           COMMON/MFLT/CU(16384).CFIL(16512).X(128).WL.NPTS.DRX.DRY
           COMPLEX CU.CFIL
     *INSERT AEROW.3
           LEVEL 2.CU
     *DELETE AEROW.9.AEROW.10
           PMX=0.
           CNT=0
           TWPI= 6.2831853
           SIGMAM = TWPI * .25
           NOM= NPTS*NPTS
     *DELETE AEROW.12.AEROW.21
           P=RANF(I)*SIGMAM
           CNT=CNT + 1
           IF(P.LE.PMX) GO TO 20
           PMX = P
   20       CONTINUE
   1     CU(I) = CU(I)*CEXP(CMPLX(0..-P))
     *INSERT AEROW.22
           WRITE(6.100)PMX.CNT
   100      FORMAT(20X.* MAX PHASE SHIFT= *.F15.7.*RADIANS*.E15.6.//)
```

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