A MANUAL TO THE MAXRAY PROGRAM LIBRARY FOR REFLECTIVE AND DISPERSIVE RAY TRACING

by

Svante Svensson and Ralf Nyholm
Uppsala University
Department of Physics
Box 530, S-751 21 Uppsala, Sweden

UUIP-1139 July.1985

UPPSALA UNIVERSITY

DEPARTMENT OF PHYSICS

A MANUAL TO THE MAXRAY PROGRAM LIBRARY FOR REFLECTIVE AND DISPERSIVE RAY TRACING

by

Svante Svensson and Ralf Nyholm
Uppsala University
Department of Physics
Box 530, S-751 21 Uppsala, Sweden

UUIP-1139 July.1985

ABSTRACT

A general ray tracing program package for reflective and dispersive X-ray optics is described. The package consists of a number of subroutines written in FORTRAN 77 code giving the necessary tools for ray tracing.
The program package is available on request from the authors.

1. INTRODUCTION

The design of X-ray and VUV optics for synchrotron radiation research is greatly facilitated by using ray tracing methods. The general ray tracing prodedure is well described in literature /1,2/ and ray tracing programs are reported from many synchrotron radiation laboratories /3,4,5/. When studying alternatives for the design of a soft X-ray monochromator for the MAX synchrotron radiation facility in Lund /6/ we however found that most existing ray tracing programs are not generally applicable to a given problem or to a certain computer system. We therefore found it convenient to make a program library consisting of a number of subroutines each of which performs a certain task necessary for the ray tracing procedure. Computer specific routines required for input or output (plotters,graphic screens etc) are left to the user. The MAXRAY program library is to be regarded as a general set of tools for raytracing.

THE MAXRAY ROUTINES ARE WRITTEN IN STANDARD FORTRAN 77 CODE USING DOUBLE PRECISION (REAL*8)

2. RAY TRACING - HOW TO MAKE IT

2.1 THE USER WRITTEN MAIN PROGRAM

A general ray tracing program consists of a MAIN program that has to be written by the user and is specific for the case to be studied. The MAIN program usually consists of three parts:

a) CREATION OF A RAY

This part of the program generates rays and thus simulates a source. This means that the program generates two sets of vectors. One set of points X and a corresponding set of DK-vectors giving the direction cosines for the ray. An arbitrary point X' on the ray is given by the parametric equation:

$$\underline{X}' = \underline{X} + S*\underline{DK}$$

where S is the distance from the point $\underline{X}'$ to the point $\underline{X}$.

The radiation source may be generated by some algorithm or by statistical simulation. In using the MAXRAY library we have applied both methods. A routine SRSOURCE (see section 4.2) is supplied in the program library. This subroutine generates a randomly simulated synchrotron radiation source for a bending magnet.

b) TRANSMISSION OF A RAY THROUGH THE SYSTEM

After generating a ray one needs to call a sequence of subroutines which transmits the ray through the given system. Different kinds of subroutines are nessecary to obtain this.

I. TRANSLATE AND ROTATE SUBROUTINES

The reflective or dispersive elements (mirrors or gratings) are described in local coordinate systems. This means that one has to transform the equation of the ray to the local system of the element before letting it act on the ray. The transformations are of two kinds:

i) A mere translation of the origo by a translation vector $\underline{TR}$. This task is performed by the subroutine TRANSLATE(X,TR).

ii) A rotation of the system around one of the coordinate axes. This is obtained in MAXRAY by calling the subroutine ROTATE(X1,X2,R1,R2).

II SUBROUTINES FOR OPTICAL ELEMENTS

Once the ray is given in the local system of an optical element one can apply one of the optical element subroutines in the MAXRAY program library to get the effect of the element. This means that the subroutine finds an intersection point of the ray with the element surface, i.e. it determines a new coordinate vector $\underline{X}$. By applying either the

general grating equation or the general reflection formula (See appendix 1.) it also finds a new DK-vector.
The optical element subroutines are described in section 3.

III SLITS AND APERTURES

In order to define slits and apertures one can use the MAXRAY subroutine PLANE(X,DK) which finds the intersection between a ray and the yz-plane. This subroutine is most commonly used to find the point of intersection with the image plane. It also serves to define slits and apertures in the system. PLANE contains the COMMON variables XMIN,XMAX and ITRAFF (see section 3.2) By setting appropriate values to these rectangular slits/apertures are defined. Upon return from the subroutine PLANE the value of the ITRAFF variable is checked (see section 3.2) and desired actions are taken (counting rays outside boundaries etc). If other shapes than rectangular ones are needed the PLANE subroutine is used to find the point of intersection with the plane of the slit/aperture and the specific boundary check is made in the MAIN program.

c) FINDING FINAL IMAGE AND OUTPUT

After the ray has passed the last optical element in the system a CALL is made to a suitable MAXRAY subroutine which finds the intersection between the ray and the image surface, i.e. it finds a final X-vector which then can be stored or tested or plotted on an output device.

In general a subroutine describing the appropriate image surface can be found among the "Optical element subroutines" described in section 3.3. Often this is simply a plane although sometimes a curved surface is required, as e.g. a cylindrical surface in the case of a Rowland type spectrometer.

2.2 HOW TO START

Before using the MAXRAY program library it is nessecary to speci-

fy the problem according to points a)-c) above i.e. one has to specify the coordinates and rotation angles for all the local coordinate systems that has to be used. Also some of the optical element subroutines (see section 3.3) require calculation of input parameters defining their geometrical shape. Sometimes it is convenient to write a short STARTUP subroutine which generates the translation vectors TR and rotation parameters R1 and R2 and other necessary parameters.

Thereafter it is a simple task to write the appropriate sequence of calls to the MAXRAY subroutines. This sequence lies in a loop in the main program generating a number of rays which simulate the source.

3. DESCRIPTION OF THE SUBROUTINES

In all the following it is implicitly assumed that X,DK and TR are vectors of dimension 3.

3.1 TRANSFORMATION OF COORDINATE SYSTEMS

3.1.1 ROTATE(X1,X2,R1,R2)

Input: X1,X2,R1,R2 Output: X1,X2

R1 and R2 are the rotation parameters. R1 and R2 have to be defined before calling ROTATE.

R1=cos(ALPHA)
R2=sin(ALPHA).

ALPHA is the desired rotation angle. If ALPHA is positive and X1, X2 and X3 are taken as a cyclic permutation of the coordinate axes in a right handed system (X,Y,Z) then calling ROTATE(X1,X2,R1,R2) implies that the coordinate system will be rotated negatively the angle ALPHA around the X3-axis. See figure 1.

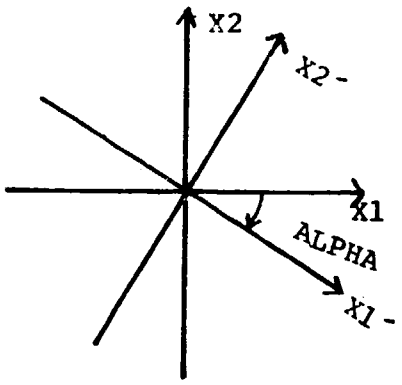


Figure 1. The coordinate systems in the ROTATE subroutine

3.1.2 TRANSLATE(X,TR)

Input: X,TR Output: X

This subroutine calculates the coordinates in a new system that has its origo shifted the vector TR relative to the old coordinate system.

3.2 RESTRICTIONS AND NORMALIZATION OF VECTORS

3.2.1 COMMON BLOCK XMIN,XMAX,ITRAFF

In the ray tracing of a system it is of great importance to sort the rays according to their different paths. For an optical element it is e.g. desirable to check wether the ray hits within some restricted area on the surface or not. In MAXRAY tests of this kind are performed using the common block variables XMIN(3), XMAX(3) and ITRAFF. Before calling a certain optical element subroutine one assigns values to XMIN and XMAX and thus defines a restriction on the coordinate values in the local system. A common such restriction is for instance a limited length or width of the element. XMIN(N) and XMAX(N) are minimum and maximum coordinate values where index N=1,2,3 refer to the X, Y and Z axes respectively. After the call is made the variable ITRAFF will be set by the subroutine to different values depending on what has happened to the ray.

Note that only two coordinates are checked by the subroutines. No check is made along the coordinate axis perpendicular to the optical element surface.

ITRAFF=0 means no intersection with the surface
ITRAFF=1 or -1 means intersection within boundaries
ITRAFF=2 or -2 means intersection outside the boundaries

A negative value of ITRAFF means that the ray has already passed the mirror. This value occurs when e.g. one is working with a virtual source.

By using the different values of ITRAFF it is possible to sort the rays (i.e. the final $\underline{X}$ vectors) and thus calculate the geometrical transmission properties of the system.

3.2.2 THE SUBROUTINE NRMLIZ(X)

Input: X Output: X

Very often it is necessary to normalize a vector when performing the calculations in the MAXRAY subroutines. This is obtained by calling the MAXRAY subroutine NRMLIZ(X) which returns a normalized vector.

3.3 THE OPTICAL ELEMENT SUBROUTINES

All the optical element subroutines are written in such a way that the x-axis of the coordinate system represents some main direction of the ray, i.e. the x-component of the direction cosine is larger than the other direction cosines. This is an assumption that is convinient when working in grazing incidence. The subroutines are however applicable also for other cases even if such a choice of system may be slightly inconvenient. The x-direction cosine ,DK(1), must however be positive in most subroutines. See appendix 3. When not explicitly stated the xy-plane is

assumed to be the tangent plane to the optical element surface in origo. In some cases we have for convenience written special subroutines where the tangent plane in origo is assumed to be the xz-plane. This is however not strictly necessary since the same action can be achieved by using an appropriate rotation (see ROTATE subroutine) of the coordinate system. In all cases of curved surfaces we only consider reflexions in a concave surface.

The user should also be warned that some unprobable but complicated cases such as multiple reflexions in the optical element or that some part of the element shadows some other part are not taken care of in these subroutines.

3.3.1 CYLINDER(X,DK,RADIUS) AND CYLINDERY(X,DK,RADIUS)

Input: X,DK,RADIUS OUTPUT: X,DK

These subroutines reflect a ray in a cylindrical surface. RADIUS is the radius of the cylinder. In CYLINDER the axis of the cylinder is parallel to the x-axis of the local system in the case of CYLINDERY the cylinder axis is parallel to the y-axis. See figure 2. below.

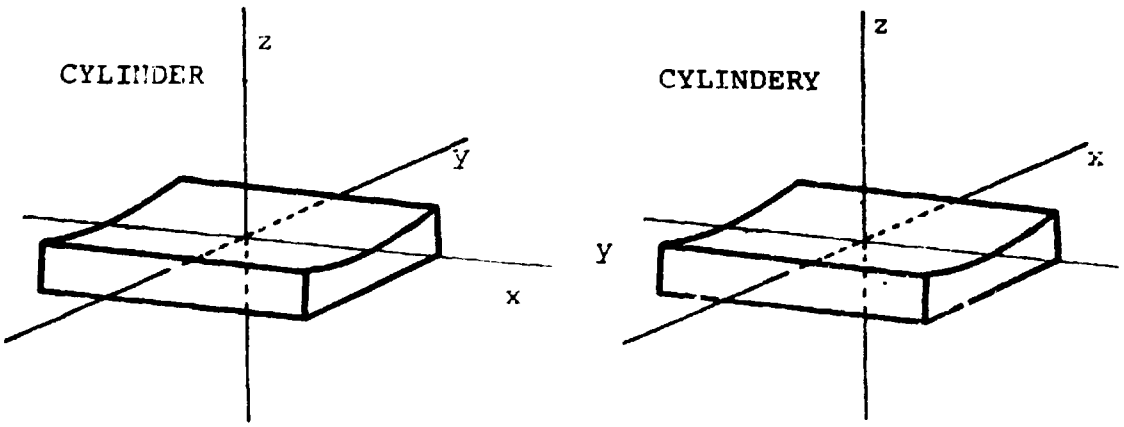


Figure 2. The coordinate systems in CYLINDER and CYLINDERY

The calculations are straightforward. The intersection of the ray with the mirror surface is found by direct solution

of the equations. A normal vector to the surface is calculated at the intersection point and the general reflection formula (See appendix 1) is applied.

3.3.2 ELLIPTIC(X,DK,RR1,RR2,FI,A,B), PLANELLY(X,DK,RR1,RR2,FI,A,B) AND PLANELLZ(X,DK,RR1,RR2,FI,A,B)

Input: X,DK,RR1,RR2,FI OUTPUT: X,DK,A,B

These subroutines calculate the reflection of a ray with an ellipsoidal mirror (ELLIPTIC) or an elliptical cylinder mirror (PLANELLY and PLANELLZ).

In the case of ELLIPTIC and PLANELLY the z-axis is perpendicular to the mirror surface in origo whereas in the case of PLANELLZ the y-axis is normal to the surface in origo.

RR1 is the distance from focus 1 to the origo of the local coordinate system of the mirror and RR2 is the distance from origo to the second focus. FI is the grazing angle of incidence in radians. See figure 3.

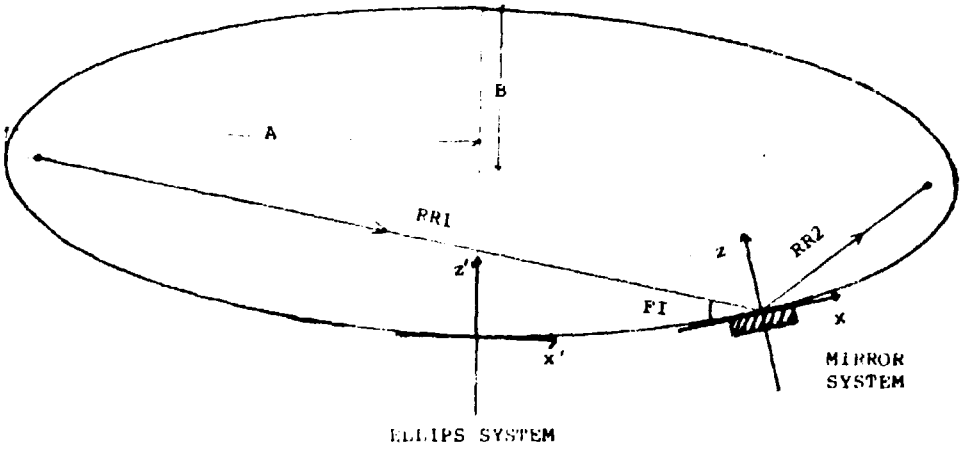


Figure 3. The coordinate systems and variables used in ELLIPTIC and PLANELLY.

A and B are the major and minor axes of the ellips. These variables are calculated by the routine and are returned tc the main program. This is done when the parameter A=0 i.e. in a normal case only the first time the subroutine is called. The minor and major axes of the ellips are not needed by the user. They are returned by the subroutine only to make a check on the calculation possible.

For computational convenience the subroutine first transforms the coordinates to a so called ellipsoid system in which the calculations are performed. At the end a transformation is made back to the local mirror system. See figure 3. The translation vector and rotation parameters necessary for this transformation are calculated only the first time the subroutine is called. These internal translations are fully transparent to the user who should only be concerned with the mirror system.

In the ellipsoid system the calculations are straightforward. First the point of intersection of the ray with the mirror surface is found by direct solution of the equations. The normal vector to the surface is found and the general reflection formula (See appendix 1) is applied.

3.3.3 PARABOLA(X,DK,D,FI,A) AND PLANPARAB(X,DK,D,FI,A)

Input: X,DK,D,FI Output: X,DK,A

These subroutines calculate the reflection of a ray with a paraboloidal mirror (PARABOLA) and a plane-parabolic mirror (PLANPARAB).

FI is the grazing angle of incidence. D is the distance from the mirror center to the focus.

A is the minimum distance from the focus to the surface of the parabola. A is calculated by the subroutine and is returned to the main program.

The user does not strictly need the parameter A . The subroutines return the value of A only in order to make checks possible. See figure 4.

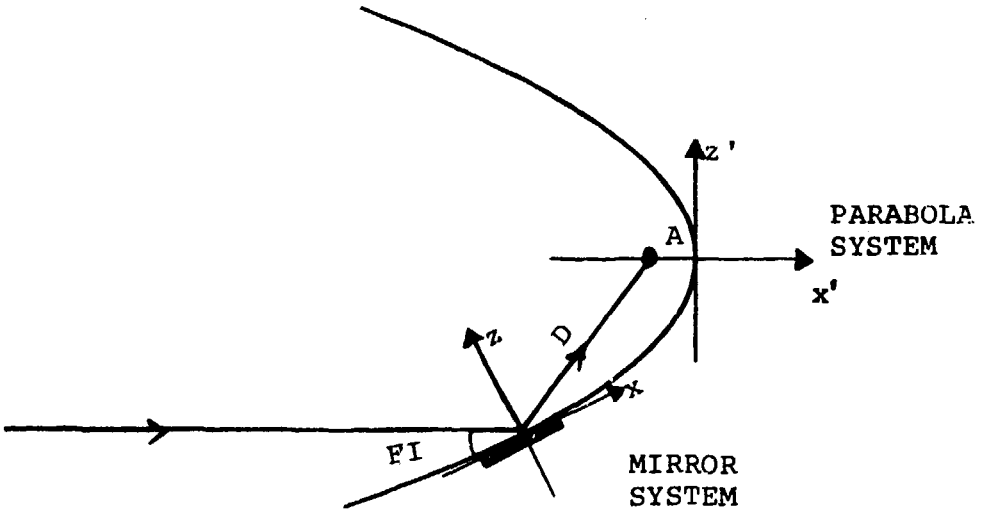


Figure 4. The coordinate system used in PARABOLA and PLANPARAB

As in the case of the elliptic mirrors the calculations are performed in the so called paraboloid coordinate system. See figure above The translation vectors and rotation parameters of this system , with respect to the local coordinate system of the mirror, are calculated when the parameter A=0 , i.e. in a normal case only the first time the subroutine is called.

The calculations are made in a manner analogous to the calculations in section 3.3.2.

3.3.4 SPHERE(X,DK,R)

INPUT: X,DK,R Output: X,DK

This subroutine reflects a ray in a spherical mirror.
The z-axis is assumed to be normal to the mirror surface at

the origo of the local coordinate system and pointing towards the center of the sphere.

R is the radius of curvature for the mirror.

The calculation proceeds in a manner analogous to the calculation in section 3.3.1.

3.3.5. TOROID(X,DK,RMAJ,RMIN)

Input: X,DK,RMAJ,RMIN *Output:* X,DK

This subroutine reflects a ray in a toroidal mirror. RMAJ and RMIN in the subroutine are defined in figure 5. The z-axis is assumed to be normal to the mirror surface at the origo of the local coordinate system. The rotation axes for generating the surface with radius RMAJ is parallel to the y-axis.

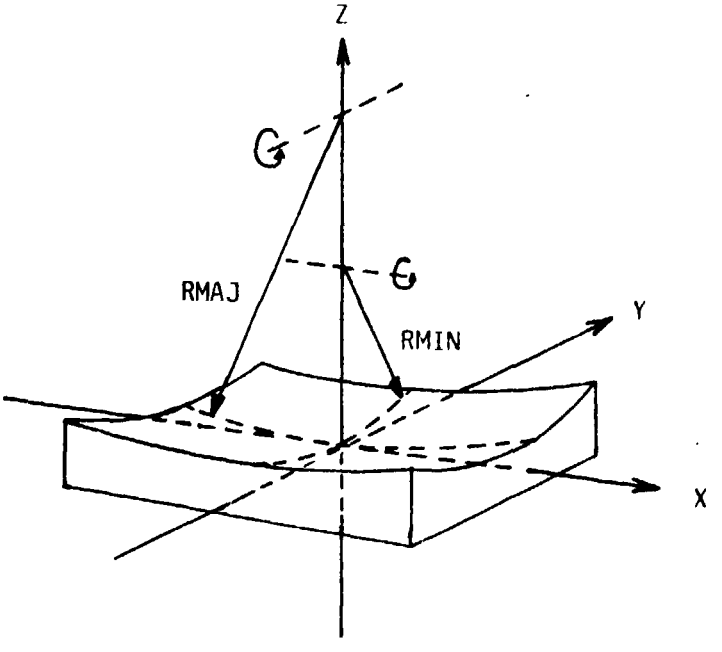


Figure 5. The coordinate system and variables in TOROID

The intersection of the ray with the mirror surface in this case has to be calculated in a Newton Raphson iteration.The gradient of the surface at the intersection point is

calculated and the general reflection formula is applied.
The starting point for the Newton Raphson iteration is taken as the intersection of the ray with the xy-plane.
The program does not use the general equation of a toroid surface (see ref.1.) but uses a restricted version in which the point of intersection having the smallest z-value is chosen. There is no restriction on RMAJ and RMIN. One may very well use a RMIN greater than RMAJ.

3.3.6 REFLECT(X,DK)

Input: X,DK Output: X,DK

This subroutine finds the point of intersection between the ray and the xy-plane and makes a subsequent reflection in this plane.

3.3.7 GRATING(X,DK,PAR)

Input: X,DK,PAR Output: X,DK

This subroutine calculates the diffraction of a ray in a plane grating. The grating lies in the xy-plane and the grooves are parallell to the y-axis. The z-axis is pointing out of the grating surface.

The intersection point with the surface is calculated and the general grating equation is taken in this point. See appendix 1.

The parameter PAR equals the wavelength multiplied with the order of diffraction and divided by the distance between sucessive grooves.

3.3.8. SPHERGRAT(X,DK,PAR,RADIUS)

Input: X,DK,PAR,RADIUS Output: X,DK

This subroutine calculates the diffraction of a ray in a spherical grating. The surface is described in the same way as in the subroutine SPHERE (Sect. 3.3.4). The z-axis of the local coordinate system is perpendicular to the surface at origo.

The parameter PAR equals the wavelength multiplied with the order of diffraction and divided by the distance between succesive grooves.

RADIUS is the radius of curvature of the spherical surface.

The intersection point of the ray with the surface is found by direct solution of the equations. After this a local coordinate system of vectors PTAK, QTAK and RTAK is constructed. The general grating equation is applied in this local system. See appendix 1.

3.3.9 TORGRAT(X,DK,RMAJ,RMIN,PAR)

Input: X,DK,RMAJ,RMIN,PAR Output: X,DK

This subroutine calculates the diffraction of a ray in a toroidal grating.

The toroidal surface is described in the same way as in the subroutine TOROID (see sect 3.3.5).

The parameter PAR equals the wavelength multiplied with the order of diffraction and divided by the distance between succesive grooves.

The intersection point is found using the same method as in TOROID. At this point a local system defined by the unit vectors PTAK, QTAK and RTAK is constructed. The general grating equation is applied in this system. See appendix 1.

3.4. INTERSECTION WITH A PLANE SURFACE

3.4.1 PLANE(X,DK)

Input: X,DK Output: X,DK

This subroutine finds the point of intersection of a ray with the yz-plane. See also section 2.1

4. SRSOURCE(X,DK,ISEED,HORANG,FWHM,UTSTR,RADIUS)

Input: ISEED,HORANG,FWHM,UTSTR,RADIUS Output: X,DK

This routine generates a simulation of a synchrotron radiation source from a bending magnet. The subroutine returns the point of origin (X) and direction (DK) of one ray. The position of the local coordinate system relative to the electron beam ins shown in figure 6.

The source volume is a section of the electron beam with a length given by the radius of the bending magnet (RADIUS) and the total horizontal angle of radiation (HORANG). The vertical cross section of the electron beam is considered circular with a current density having a Gaussian distribution with full width at half maximum given by the parameter UTSTR.

The source point (X) is chosen within this source volume using a random distribution sampling the current density.

The direction vector of the ray (DK) lies in a vertical plane tangential to the electron trajectory and intersecting the source point. The angle between DK and the horizontal plane is chosen randomly with a Gaussian distribution having a full width at half maximum given by the parameter FWHM.

ISEED is the argument for the random number generator in VAX 11 FORTRAN 77.

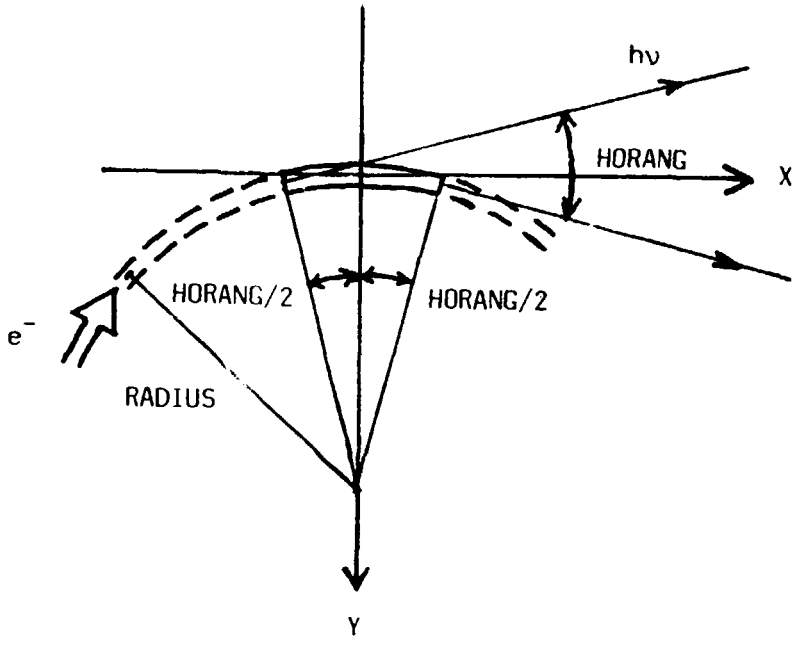


Figure 6. The coordinate system in SRSOURCE

The SRSOURCE subroutine is well suited for cases where the extension of the source is critical to the properties of the final image although one has to keep in mind that the circular cross section and the Gaussian ditribution of the current density are only approximately correct.

The simulation of the direction is not strictly correct since each source point emits radiation distributed around the tangent both vertically and horizontally. A full treatment of the angular distribution would however be too time consuming (see equation II.29 in ref. 7). Commonly the angular intensity distribution is given only in the vertical direction by integrating over the horizontal angle (see equation II.34 in ref. 7). At photon energies close to and higher than the critical energy this vertical intensity distribution is approximately Gaussian. This is the reason behind the particular choice of angular distribution in SRSOURCE. We have found that using the proper value of FWHM (depending on photon energy) one obtains correct values for the transmission of an optical system by simply counting the number of rays passing through the system.

References

1. G.H.Spencer and M.V.R.K.Murty, J.Opt.Soc.Am.,41, 630 (1962)

2. S.O.Kastner and W.M.Neupert, J.Opt.Soc.Am.,53, 1180 (1963)

3. D.Hubbard, E.Pantos, Nucl.Instr. and Meth.,208,319 (1983)

4. D.E.Aspnes and S.M.Kelso, Nucl.Instr.and Meth.,195,175 (1982)

5. B.Lai and F.Cerrina in "Int.Conf. on X-ray and VUV Synchrotron Radiation Instrumentation",Stanford, 1985; to be publ. in Nucl.Instr.and Meth.

6. R.Nyholm,S.Svensson,J.Nordgren and A.Flodström, to be publ. in Nucl.Instr. and Meth.

7. J.Schwinger, Phys. Rev. 75, 1912 (1949).

APPENDIX 1

The general reflection and general grating equation

The general grating equation /1/ is given by:

$$\hat{r} \times \hat{b} = - \hat{a} \times \hat{r} - (m\lambda/d)\, \hat{q}$$

where $\hat{p}$, $\hat{q}$, $\hat{r}$ are unit vectors defining a coordinate system. See figure 7. below.

The grooves are assumed to be parallell to the $\hat{q}$ direction.
$\hat{a}$ is a unit vector parallell to the incidence direction of the ray.
$\hat{b}$ is a unit vector parallell to the direction of the diffracted ray.

m is the order of diffraction.
λ is the wavelength of the radiation.
d is the distance between the grooves.

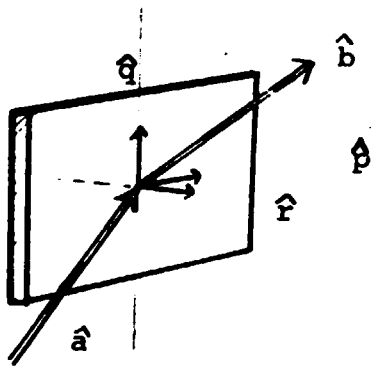


Figure 7. The coordinate system used in the general grating equation.

If the vector $\hat{a}$ is known the unit vector $\hat{b}$ can be calculated as:

$$\hat{b} = \hat{a} - (m\lambda/d)\, \hat{p} + Q\, \hat{r}$$

where $Q = - \hat{a}\cdot\hat{r} + \left((\hat{a}\cdot\hat{r})^2 + (2m\lambda/d)(\hat{a}\cdot\hat{p}) - (m\lambda/d)^2 \right)^{1/2}$

The general reflection formula is easily found from the

expressions above by considering the case m=0. One then finds:

$$\hat{b} = \hat{a} - 2\,(\hat{a}\cdot\hat{r})\hat{r}$$

In this case it is not necessary to know all the axes of the coordinate system. The reflection is determined only from the surface normal vector r .

APPENDIX 2

Fortran 77 code for the subroutines

```
      SUBROUTINE CYLINDER(X,DK,RADIUS)

***   THE ROUTINE CALCULATES THE REFLECTION OF A RAY
***   GIVEN BY X AND DK WITH THE CYLINDER:
****
****  Y**2 + (Z-RADIUS)**2 = RADIUS**2
****
***   THIS CYLINDER MAKES A TANGENT TO THE X-AXIS.
***   A NEW X IS CALCULATED FROM THE INTERSECTION POINT.
***   A NEW DK IS CALCULATED BY FIRST CALCULATING THE SURFACE
***   NORMAL DN AND THEN APPLYING THE GENERAL EQUATION FOR
***   REFLECTION
***   THE X-DIRECTION COSINE (DK(1)) MUST BE POSITIVE

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3),DN(3)

      P=DK(2)**2+DK(3)**2
      Q=X(2)*DK(2)+X(3)*DK(3)-RADIUS*DK(3)
      R=X(2)**2+X(3)**2-2*X(3)*RADIUS

      DET=Q**2-P*R

      IF ((DET.LT.0).OR.(P.EQ.0)) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=(-1*Q+DSQRT(DET))/P

      ITRAFF=1
      DO 20 I=1,3
 20   X(I)=X(I)+DK(I)*S

      ITRAFF=1
      DO 25 I=1,2
 25   IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
      IF (S.LT.0) ITRAFF=-ITRAFF

****  CALCULATE THE SURFACE NORMAL DN

      DN(1)=0.
      DN(2)=X(2)
      DN(3)=X(3)-RADIUS
      CALL NRMLIZ(DN)

****  THE SCALAR PRODUCT OF DK AND DN

      PROD=0
```

```
      DO 30 I=1,3
30    PROD=PROD+DK(I)*DN(I)

****  NEW DK IS OLD DK -2*PROD*DN

      DO 40 I=1,3
40    DK(I)=DK(I)-2*PROD*DN(I)

      RETURN
      END
```

```
      SUBROUTINE CYLINDERY(X,DK,RADIUS)

***   THIS ROUTINE CALCULATES THE REFLECTION OF A
***   RAY GIVEN BY X AND DK WITH THE CYLINDER:
****
****  X**2 + (Z-RADIUS)**2 = RADIUS**2
****
***   THIS CYLINDER MAKES A TANGENT WITH THE Y-AXIS.
***   A NEW X IS CALCULATED FROM THE POINT OF INTERSECTION.
***   A NEW DK IS CALCULATED BY FIRST CALCULATING THE SURFACE
***   NORMAL DN AND THEN APPLYING THE GENERAL EQUATION FOR
***   REFLECTION.
***   THE X-DIRECTION COSINE (DK(1)) MUST BE POSITIVE

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3),DN(3)

      P=DK(1)**2+DK(3)**2
      Q=X(1)*DK(1)+X(3)*DK(3)-RADIUS*DK(3)
      R=X(1)**2+X(3)**2-2*X(3)*RADIUS

      DET=Q**2-P*R

      IF ((DET.LT.0).OR.(P.EQ.0)) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=(-1*Q+DSQRT(DET))/P

      DO 20 I=1,3
 20   X(I)=X(I)+DK(I)*S

      ITRAFF=1
      DO 25 I=1,2
 25   IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
      IF (S.LT.0) ITRAFF=-ITRAFF

****  CALCULATE THE NORMAL VECTOR DN

      DN(1)=X(1)
      DN(2)=0.
      DN(3)=X(3)-RADIUS
      CALL NRMLIZ(DN)

****  CALCULATE THE SCALAR PRODUCT OF DK AND DN

      PROD=0
      DO 30 I=1,3
```

```
 30      PROD=PROD+DK(I)*DN(I)

****     NEW DK CALCULATED AS OLD DK- 2*PROD*DN

         DO 40 I=1,3
 40      DK(I)=DK(I)-2*PROD*DN(I)

         RETURN
         END
```

```
      SUBROUTINE ELLIPTIC(X,DK,RR1,RR2,FI,A,B)

****  THIS ROUTINE CALCULATES THE REFLECTION OF A RAY
****  GIVEN BY X AND DK WITH THE ELLIPSOID:

****  X**2/A**2  + (Y**2+(Z-B)**2)/B**2 = 1

****  (THE ELLIPSOID ABOVE IS GIVEN IN THE ELLIPSOID SYSTEM.
****  SEE DOCUMENTATION OF MAXRAY SUBROUTINES)

***   RR1=DISTANCE FROM FOCUS 1 TO THE MIRROR CENTER
***   RR2=DISTANCE FROM FOCUS 2 TO THE MIRROR CENTER
***   FI=GRAZING ANGLE IN RADIANS

****  A NEW X IS CALCULATED FROM THE POINT OF INTERSECTION
****  AND A NEW DK IS CALCULATED BY FIRST CALCULATING THE
****  SURFACE NORMAL VECTOR AND THEN APPLYING THE GENERAL
****  EQUATION FOR REFLECTION.
***   THE X-DIRECTION COSINE (DK(1)) MUST BE POSITIVE

*************************************************************
****  NOTICE THAT IN THE CALL TO ELLIPTIC IT IS ASSUMED THAT
****  THE X AND DK VECTORS ARE GIVEN IN THE MIRROR SYSTEM!!!
*************************************************************

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),XO(3),DK(3),DK2(3),DN(3),GRADYT(3),XYTA(3)

**********************************************************************
***   HERE FOLLOWS A ROUTINE TO CALCULATE THE AXES
***   OF THE ELLIPSOID AND THE TRANSLATION VECTOR AND THE ROTATION
***   PARAMETERS BETWEEN THE ELLIPSOID SYSTEM AND THE MIRROR
***   SYSTEM. THIS CALCULATION IS DONE ONLY IF THE PARAMETER
***   'A' IS ZERO.
**********************************************************************

      IF(A.EQ.0) THEN

      C=SQRT(RR1**2+RR2**2+2*RR1*RR2*COS(2*FI))/2.
      A=(RR1+RR2)/2.
      B=SQRT(A*A-C*C)
      THETA1=ASIN(RR2*SIN(2*FI)/C/2)
      A2=A**2
      B2=B**2

***   XYTA GIVES THE TRANSLATION VECTOR FROM THE ELLIPSOID
***   SYSTEM TO THE MIRROR SYSTEM

      XYTA(1)=-C+RR1*COS(THETA1)
```

```
      XYTA(3)=B-RR1*SIN(THETA1)

***   GRADYT GIVES THE NORMAL TO THE ELLIPSOID SURFACE AT THE CENTER
***   OF THE MIRROR

      GRADYT(1)=-2*XYTA(1)/A2
      GRADYT(2)=0.
      GRADYT(3)=-2*(XYTA(3)-B)/B2
      CALL NRMLIZ(GRADYT)

****  R1 AND R2 ARE THE ROTATION PARAMETERS BETWEEN
****  THE ELLIPSOID AND THE MIRROR SYSTEM.

      R1=GRADYT(3)
      R2=GRADYT(1)
      ENDIF

***   HERE FOLLOWS THE CALCULATION OF REFLECTION WITH THE ELLIPSOID
***   THIS IS ALL MADE IN THE SYSTEM OF THE ELLIPSOID.

****  FIRST MAKE A TRANSFORMATION FROM THE MIRROR SYSTEM TO THE
****  SYSTEM OF THE ELLIPSOID

      CALL ROTATE(X(3),X(1),R1,R2)
      CALL ROTATE(DK(3),DK(1),R1,R2)

      DO I=1,3
      XYTA(I)=-XYTA(I)
      ENDDO
      CALL TRANSLATE(X,XYTA)
      DO I=1,3
      XYTA(I)=-XYTA(I)
      ENDDO

****  REFLECTION WITH THE ELLIPSOID

      DO 10 I=1,3
 10   DK2(I)=DK(I)**2

      P=DK2(1)/A2+(DK2(2)+DK2(3))/B2
      Q=DK(1)*X(1)/A2+DK(3)*X(3)/B2+DK(2)*X(2)/B2-DK(3)/B
      R=X(1)**2/A2+((X(2)**2+X(3)**2)/B-2*X(3))/B

      DET=Q**2-P*R

      IF (DET.LT.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=(-1*Q+DSQRT(DET))/P

      DO 20 I=1,3
 20   X(I)=X(I)+DK(I)*S

****  THE NORMAL VECTOR DN IS CALCULATED
```

```
      DN(1)=2*X(1)/A2
      DN(2)=2*X(2)/B2
      DN(3)=2*(X(3)-B)/B2
      CALL NRMLIZ(DN)

****  THE  SCALAR PRODUCT OF DK AND DN IS CALCULATED

      PROD=0
      DO 30 I=1,3
 30   PROD=PROD+DK(I)*DN(I)

****  THE NEW DK IS CALCULATED AS OLD DK -2*PROD*DN

      DO 40 I=1,3
 40   DK(I)=DK(I)-2*PROD*DN(I)

****  NOW WE TRANSFORM BACK TO THE MIRROR SYSTEM

      CALL TRANSLATE(X,XYTA)
      R2=-R2
      CALL ROTATE(X(3),X(1),R1,R2)
      CALL ROTATE(DK(3),DK(1),R1,R2)
      R2=-R2

****  TEST IF THE RAY HITS THE MIRROR

      ITRAFF=1
      DO I=1,2
      IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
      ENDDO
      IF (S.LT.0) ITRAFF=-ITRAFF

      RETURN
      END
```

```
      SUBROUTINE GRATING(X,DK,PAR)

***   THE ROUTINE CALCULATES A NEW X AND DK ACCORDING TO THE
***   GENERAL EQUATION FOR A PLANE GRATING. SEE SPENCER AND MURTY
***   J.OPT.SOC. OF AM. VOL.52 NUMBER 6 P.672 FF(1962)
***   PAR=M*LAMBDA/D IS DEFINED IN MAIN PROGRAM
***   OBSERVE THE ORIENTATION OF THE COORDINATE SYSTEM
***   THE X AXIS IS ORTHOGONAL TO THE GROOVES AND LIES IN THE
***   PLANE OF THE GRATING. THE Z-AXIS IS ORTHOGONAL TO THE
***   PLANE OF THE GRATING.
***   THE Z-DIRECTION COSINE (DK(3)) MUST BE NEGATIVE

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),XO(3),DK(3)

      IF(DK(3).EQ.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=-X(3)/DK(3)
      X(1)=X(1)+DK(1)*S
      X(2)=X(2)+DK(2)*S
      X(3)=0

      ITRAFF=1
      DO 20 I=1,2
   20 IF((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
      IF(S.LT.0) ITRAFF=-ITRAFF

***   GENERAL GRATING EQUATION

      Q=-DK(3)+SQRT(DK(3)**2+PAR*(2*DK(1)-PAR))
      DK(1)=DK(1)-PAR
      DK(2)=DK(2)
      DK(3)=DK(3)+Q

      RETURN
      END
```

```
      SUBROUTINE NRMLIZ(X)

****  THIS SUBROUTINE MAKES NORMALIZATION OF A VECTOR

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(1:3)

      XB=0

      DO  I=1,3
      XB=XB+X(I)*X(I)
      ENDDO

      XB=SQRT(XB)

      DO I=1,3
      X(I)=X(I)/XB
      ENDDO

      RETURN
      END
```

```
      SUBROUTINE PARABOLA(X,DK,D,FI,A)

****  THIS SUBROUTINE CALCULATES THE REFLECTION OF A
****  RAY GIVEN BY X AND DK WITH THE PARABOLOID:
****
****  4*A*X + Y**2 + Z**2 = 0

***   (THIS IS THE EQUATION IN THE PARABOLOID SYSTEM)

****  D=DISTANCE FROM MIRROR CENTER TO FOCUS
****  FI = GRAZING ANGLE IN RADIANS
****
****  THE PARABOLOID MAKES A TANGENT TO THE YZ PLAN IN ORIGO.
****  A NEW X IS CALCULATED AS THE POINT OF INTERSECTION.
****  A NEW DK IS CALCULATED BY FIRST CALCULATING THE SURFACE
****  NORMAL DN AND THEN APPLYING THE GENERAL EQUATION
****  FOR REFLECTION.
****  THE X-DIRECTION COSINE MUST BE POSITIVE

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

*************************************************************
****  NOTICE THAT IN THE CALL TO PARABOLA IT IS ASSUMED THAT
****  X AND DK ARE GIVEN IN THE MIRROR SYSTEM
*************************************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3),DN(3),XYTA(3)

*************************************************************************
***   HERE FOLLOWS A ROUTINE TO CALCULATE THE AXIS OF THE PARABOLOID
***   AND THE TRANSLATION  VECTOR AND ROTATION PARAMETERS BETWEEN THE
***   PARABOLOID SYSTEM AND THE MIRROR SYSTEM. THIS CALCULATION IS
***   DONE ONLY IF THE PARAMETER 'A' IS ZERO.
*************************************************************************

***   XYTA=THE TRANSLATION VECTOR FROM THE  PARABOLOID SYSTEM ORIGO
***   TO THE MIRROR SYSTEM ORIGO
***
***   R1 AND R2 DEFINES THE ROTATION PARAMETERS TO ROTATE FROM THE
***   MIRROR SYSTEM TO THE PARABOLOID SYSTEM

      IF(A.EQ.0) THEN

      A=D*SIN(FI)**2
      XYTA(1)=A-D
      XYTA(2)=0
      XYTA(3)=-D*SIN(2*FI)

      R1=COS(FI)
      R2=SIN(FI)
```

```
      ENDIF

*****************************************************************************
***
***     HERE FOLLOWS THE CALCULATION OF REFLECTION WITH THE PARABOLOID
***     THIS IS ALL MADE IN THE PARABOLOID SYSTEM

***     TRANSFORMATION FROM THE MIRROR SYSTEM TO THE PARABOLOID SYSTEM

      R2=-R2
      CALL ROTATE(X(3),X(1),R1,R2)
      CALL ROTATE(DK(3),DK(1),R1,R2)
      R2=-R2

      DO I=1,3
      XYTA(I)=-XYTA(I)
      ENDDO
      CALL TRANSLATE(X,XYTA)
      DO I=1,3
      XYTA(I)=-XYTA(I)
      ENDDO

***     CALCULATE THE POINT OF INTERSECTION

      P=DK(2)**2+DK(3)**2
      Q=(2*A*DK(1)+X(2)*DK(2)+X(3)*DK(3))
      R=4*A*X(1)+X(2)**2+X(3)**2
      DET=Q**2-P*R
      SER=P*R/Q**2

      IF (DET.LT.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      IF (P.EQ.0) THEN
          S=-R*.5/Q
      ELSEIF (ABS(SER).LT.1.E-4) THEN
          S=0.5*Q/P*SER*(-1-SER/4-SER**2/8)
      ELSE
          S=(-1*Q+SQRT(DET))/P
      ENDIF

      DO 20 I=2,3
 20   X(I)=X(I)+DK(I)*S
      X(1)=-(X(2)**2+X(3)**2)/4/A

****  THE NORMAL VECTOR DN

      DN(1)=-4*A
      DN(2)=-2*X(2)
      DN(3)=-2*X(3)
      CALL NRMLIZ(DN)

****  THE SCALAR PRODUCT OF DN AND DK
```

```
      PROD=0
      DO 30 I=1,3
 30   PROD=PROD+DK(I)*DN(I)

****  NEW DK = OLD DK-2*PROD*DN

      DO 40 I=1,3
 40   DK(I)=DK(I)-2*PROD*DN(I)

***   TRANSFORMATION TO THE MIRROR SYSTEM

      CALL TRANSLATE(X,XYTA)
      CALL ROTATE(X(3),X(1),R1,R2)
      CALL ROTATE(DK(3),DK(1),R1,R2)

***   TEST IF THE RAY HAS HIT THE MIRROR

      ITRAFF=1
      DO 21 I=1,2
 21   IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
      IF (S.LT.0) ITRAFF=-ITRAFF

      RETURN
      END
```

```
      SUBROUTINE PLANE(X,DK)

***   THIS ROUTINE CALCULATES THE POINT OF INTERSECTION BETWEEN A
***   RAY GIVEN BY X AND DK AND THE PLANE X=0

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3)

      IF(DK(1).EQ.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=-X(1)/DK(1)
      X(2)=X(2)+DK(2)*S
      X(3)=X(3)+DK(3)*S
      X(1)=0

      ITRAFF=1
      DO I=2,3
      IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
      ENDDO
      IF(S.LT.0) ITRAFF=-ITRAFF

      RETURN
      END
```

```
      SUBROUTINE PLANELLY(X,DK,RR1,RR2,FI,A,B)

****  THIS ROUTINE CALCULATES THE REFLECTION OF A RAY
****  GIVEN BY X AND DK WITH THE PLANE-ELLIPS (ELLIPTICAL CYLINDER):

****  X**2/A**2  + (Z-B)**2/B**2 = 1

****  THIS ELLIPS MAKES A TANGENT TO THE Y-AXIS IN ORIGO.
****  A NEW X IS CALCULATED FROM THE POINT OF INTERSECTION.
****  A NEW DK IS CALCULATED BY FIRST CALCULATING THE SURFACE
****  NORMAL VECTOR AND THEN APPLYING THE GENERAL EQUATION
****  FOR REFLECTION.
****  THE X-DIRECTION COSINE (DK(1)) MUST BE POSITIVE

****  RR1=THE DISTANCE FROM FOCUS 1 TO THE MIRROR CENTER
****  RR2=THE DISTANCE FROM FOCUS 2 TO THE MIRROR CENTER
****  FI=GRAZING ANGLE IN RADIANS

*****************************************************************
****  NOTICE THAT IN THE CALL TO PLANELLY IT IS ASSUMED THAT
****  THE X AND DK VECTORS ARE GIVEN IN THE MIRROR SYSTEM!!!
*****************************************************************

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),XO(3),DK(3),DK2(3),DN(3),GRADYT(3),XYTA(3)

*****************************************************************
***   HERE FOLLOWS A ROUTINE TO CALCULATE THE AXES
***   OF THE ELLIPS AND THE TRANSLATION VECTOR AND THE ROTATION
***   PARAMETERS BETWEEN THE ELLIPS SYSTEM AND THE MIRROR
***   SYSTEM. THIS CALCULATION IS DONE ONLY IF THE PARAMETER
***   'A' IS ZERO.
*****************************************************************

      IF(A.EQ.0) THEN

      FI2=FI*2.
      C=SQRT(RR1**2+RR2**2+2.*RR1*RR2*COS(FI2))/2.
      A=(RR1+RR2)/2.
      B=SQRT(A*A-C*C)
      THETA1=ASIN(RR2*SIN(FI2)/C/2.)
      A2=A**2
      B2=B**2

***   XYTA GIVES THE TRANSLATION VECTOR FROM THE ELLIPSOID
***   SYSTEM TO THE MIRROR SYSTEM

      XYTA(1)=-C+RR1*COS(THETA1)
      XYTA(2)=0.
```

```
      XYTA(3)=B-RR1*SIN(THETA1)

***   GRADYT GIVES THE NORMAL TO THE ELLIPSOID SURFACE AT THE MIDDLE
***   OF THE MIRROR

      GRADYT(1)=-2*XYTA(1)/A2
      GRADYT(2)=0.
      GRADYT(3)=-2*(XYTA(3)-B)/B2
      CALL NRMLIZ(GRADYT)

****  ROTYT GIVES THE ROTATION ANGLE BETWEEN THE ELLIPSOID AND THE
****  MIRROR SYSTEM.

      R1=GRADYT(3)
      R2=GRADYT(1)

      ENDIF

**********************************************************************
***
***   HERE FOLLOWS THE CALCULATION OF REFLECTION WITH THE PLANE-
***   ELLIPS. THIS IS ALL MADE IN THE ELLIPS SYSTEM
***
**********************************************************************

****  TRANSFORMATION TO THE SYSTEM OF THE ELLIPS

      CALL ROTATE(X(3),X(1),R1,R2)
      CALL ROTATE(DK(3),DK(1),R1,R2)

      DO I=1,3
      XYTA(I)=-XYTA(I)
      ENDDO
      CALL TRANSLATE(X,XYTA)

      DO I=1,3
      XYTA(I)=-XYTA(I)
      ENDDO

****  REFLECTION IN THE ELLIPS

      DO 10 I=1,3
 10   DK2(I)=DK(I)**2

      P=DK2(1)/A2+DK2(3)/B2
      Q=DK(1)*X(1)/A2+DK(3)*X(3)/B2-DK(3)/B
      R=X(1)**2/A2+(X(3)**2/B-2*X(3))/B

      DET=Q**2-P*R

      IF (DET.LT.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=(-Q+SQRT(DET))/P
      DO 20 I=1,3
```

```
 20     X(I)=X(I)+DK(I)*S

****    THE NORMAL VECTOR DN IS CALCULATED

        DN(1)=2*X(1)/A2
        DN(2)=0.
        DN(3)=2*(X(3)-B)/B2
        CALL NRMLIZ(DN)

****    THE  SCALAR PRODUCT OF DK AND DN IS CALCULATED

        PROD=0
        DO 30 I=1,3
 30     PROD=PROD+DK(I)*DN(I)

****    THE NEW DK IS CALCULATED AS OLD DK - 2*PROD*DN

        DO 40 I=1,3
 40     DK(I)=DK(I)-2*PROD*DN(I)

****    TRANSFORM BACK TO THE MIRROR SYSTEM

        CALL TRANSLATE(X,XYTA)
        R2=-R2
        CALL ROTATE(X(3),X(1),R1,R2)
        CALL ROTATE(DK(3),DK(1),R1,R2)
        R2=-R2

****    TEST IF THE RAY HITS THE MIRROR

        ITRAFF=1
        DO I=1,2
        IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
        ENDDO
        IF(S.LT.0) ITRAFF=-ITRAFF

        RETURN
        END
```

```
      SUBROUTINE PLANELLZ(X,DK,RR1,RR2,FI,A,B)

****  THIS ROUTINE CALCULATES THE REFLECTION OF A RAY GIVEN
****  BY X AND DK WITH THE PLANE-ELLIPS (ELLIPTICAL CYLINDER):

****  X**2/A**2  + (Y-B)**2/B**2 = 1

****  THIS ELLIPS MAKES A TANGENT TO THE Z-AXIS IN ORIGO.
****  A NEW X IS CALCULATED FROM THE POINT OF INTERSECTION.
****  A NEW DK IS CALCULATED BY FIRST CALCULATING THE SURFACE
****  NORMAL VECTOR DN AND THEN APPLYING THE GENERAL EQUATION
****  FOR REFLECTION.
***   THE X-DIRECTION COSINE (DK(1)) MUST BE POSITIVE

***   RR1=DISTANCE FROM FOCUS 1 TO MIRROR CENTER
***   RR2=DISTANCE FROM FOCUS 2 TO MIRROR CENTER
***   FI=GRAZING ANGLE IN RADIANS

******************************  ******************************
****  NOTICE THAT IN THE CALL TO PLANELLZ IT IS ASSUMED THAT
****  THE X AND DK VECTORS ARE GIVEN IN THE MIRROR SYSTEM!!!
**************************************************************

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),XO(3),DK(3),DK2(3),DN(3),GRADYT(3),XYTA(3)

******************************************************************
***   HERE FOLLOWS A ROUTINE TO CALCULATE THE AXES
***   OF THE ELLIPS AND THE TRANSLATION VECTOR AND THE ROTATION
***   PARAMETERS BETWEEN THE ELLIPS SYSTEM AND THE MIRROR
***   SYSTEM. THIS CALCULATION IS DONE ONLY IF THE PARAMETER
***   'A' IS ZERO.
******************************************************************

      IF(A.EQ.0) THEN

      FI2=FI*2.
      C=SQRT(RR1**2+RR2**2+2*RR1*RR2*COS(FI2))/2.
      A=(RR1+RR2)/2.
      B=SQRT(A*A-C*C)
      THETA1=ASIN(RR2*SIN(FI2)/C/2)
      A2=A**2
      B2=B**2

***   XYTA GIVES THE TRANSLATION VECTOR FROM THE ELLIPS
***   SYSTEM TO THE MIRROR SYSTEM

      XYTA(1)=-C+RR1*COS(THETA1)
      XYTA(2)=B-RR1*SIN(THETA1)
```

```
***     GRADYT GIVES THE NORMAL TO THE ELLIPS SURFACE AT THE MIDDLE
***     OF THE MIRROR

        GRADYT(1)=-2*XYTA(1)/A2
        GRADYT(3)=0.
        GRADYT(2)=-2*(XYTA(2)-B)/B2
        CALL NRMLIZ(GRADYT)

***     R1 AND R2 ARE THE ROTATION PARAMETERS BETWEEN THE
***     ELLIPS AND THE MIRROR SYSTEM.

        R1=GRADYT(2)
        R2=GRADYT(1)

        ENDIF

****************************************************************
***     HERE FOLLOWS THE CALCULATION OF REFLECTION IN THE PLANE-
***     ELLIPS. THIS IS ALL DONE IN THE SYSTEM OF THE ELLIPS
****************************************************************

****    TRANSFORM TO THE SYSTEM OF THE ELLIPS

        R2=-R2
        CALL ROTATE(X(1),X(2),R1,R2)
        CALL ROTATE(DK(1),DK(2),R1,R2)
        R2=-R2

        DO I=1,3
        XYTA(I)=-XYTA(I)
        ENDDO
        CALL TRANSLATE(X,XYTA)
        DO I=1,3
        XYTA(I)=-XYTA(I)
        ENDDO

****    REFLECTION IN THE ELLIPS

        DO 10 I=1,3
 10     DK2(I)=DK(I)**2

        P=DK2(1)/A2+DK2(2)/B2
        Q=DK(1)*X(1)/A2+DK(2)*X(2)/B2-DK(2)/B
        R=X(1)**2/A2+(X(2)**2/B-2*X(2))/B

        DET=Q**2-P*R

        IF (DET.LT.0) THEN
        ITRAFF=0
        RETURN
        ENDIF

        S=(-1*Q+SQRT(DET))/P
        DO 20 I=1,3
  20    X(I)=X(I)+DK(I)*S
```

```
****    THE NORMAL VECTOR DN IS CALCULATED

        DN(1)=2*X(1)/A2
        DN(3)=0.
        DN(2)=2*(X(2)-B)/B2
        CALL NRMLIZ(DN)

****    THE  SCALAR PRODUCT OF DK AND DN IS CALCULATED

        PROD=0
        DO 30 I=1,3
 30     PROD=PROD+DK(I)*DN(I)

****    THE NEW DK IS CALCULATED AS OLD DK - 2*PROD*DN

        DO 40 I=1,3
 40     DK(I)=DK(I)-2*PROD*DN(I)

****    TRANSFORM BACK TO THE MIRROR SYSTEM

        CALL TRANSLATE(X,XYTA)
        CALL ROTATE(X(1),X(2),R1,R2)
        CALL ROTATE(DK(1),DK(2),R1,R2)

****    TEST IF THE RAY HITS THE MIRROR

        ITRAFF=1
        DO I=1,3,2
        IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
        ENDDO
        IF(S.LT.0) ITRAFF=-ITRAFF

         RETURN
         END
```

```
      SUBROUTINE PLANPARAB(X,DK,D,FI,A)

****  THIS SUBROUTINE CALCULATES THE REFLECTION OF
****  A RAY GIVEN BY X AND DK WITH THE PARABOLA:
****
****  4*A*X +  Z**2 = 0

****  THE PARABOLA MAKES A TANGENT TO THE YZ PLANE IN ORIGO.
****  A NEW X IS CALCULATED AS THE POINT OF INTERSECTION.
****  A NEW DK IS CALCULATED BY FIRST CALCULATING THE SURFACE
****  NORMAL DN AND THEN APPLYING THE GENERAL EQUATION
****  FOR REFLECTION.
****  THE X-DIRECTION COSINE (DK(1)) MUST BE POSITIVE

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

***   FI=THE GRAZING ANGLE FOR THE MIRROR
***   D = THE DISTANCE FROM THE FOCUS TO THE MIRROR

**********************************************************************
****  NOTICE THAT IN THE CALL TO PLANPARAB IT IS ASSUMED THAT
****  THE X AND DK VECTORS ARE GIVEN IN THE MIRROR SYSTEM!!!
**********************************************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3),DN(3),XYTA(3)

**********************************************************************
***   HERE FOLLOWS A ROUTINE TO CLACULATE THE ROTATION AND TRANS-
***   LATION PARAMETERS BETWEEN THE SYSTEM OF THE PARABOLA AND
***   THE SYSTEM OF THE MIRROR. THIS CALCULATION IS DONE ONLY
***   IF THE PARAMETER 'A' IS ZERO.
**********************************************************************

***   XYTA=THE TRANSLATION VECTOR FROM THE  PARABOLOID SYSTEM ORIGO
***   TO THE MIRROR SYSTEM ORIGO
***
***   R1 AND R2 DEFINES THE ROTATION PARAMETERS TO ROTATE FROM THE
***   MIRROR SYSTEM TO THE PARABOLOID SYSTEM

***   CALCULATE THE TRANSLATION VECTOR XYTA

      IF(A.EQ.0) THEN

      A=D*SIN(FI)**2
      XYTA(1)=A-D
      XYTA(2)=0
      XYTA(3)=-D*SIN(2*FI)

***   THE ROTATION PARAMETERS R1 AND R2 ARE DEFINED ACCORDING TO A
***   POSITIVE ROTATION AROUND THE Y AXIS
```

```
      R1=COS(FI)
      R2=-SIN(FI)

      ENDIF

***   TRANSFORM TO THE PARABOLOID SYSTEM

      CALL ROTATE(X(3),X(1),R1,R2)
      CALL ROTATE(DK(3),DK(1),R1,R2)

      DO I=1,3
      XYTA(I)=-XYTA(I)
      ENDDO
      CALL TRANSLATE(X,XYTA)
      DO I=1,3
      XYTA(I)=-XYTA(I)
      ENDDO

***   CALCULATE THE POINT OF INTERSECTION

      P=DK(3)**2
      Q=(2*A*DK(1)+X(3)*DK(3))
      R=4*A*X(1)+X(3)**2
      DET=Q**2-P*R
      SER=P*R/Q**2

      IF (DET.LT.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      IF (P.EQ.0) THEN
         S=-R*.5/Q
      ELSEIF (ABS(SER).LT.1.E-4) THEN
         S=0.5*Q/P*SER*(-1-SER/4-SER**2/8)
      ELSE
         S=(-1*Q+SQRT(DET))/P
      ENDIF

      DO 20 I=2,3
 20   X(I)=X(I)+DK(I)*S
      X(1)=-X(3)**2/4/A

****  THE NORMAL VECTOR DN

      DN(1)=-4*A
      DN(2)=0.
      DN(3)=-2*X(3)
      CALL NRMLIZ(DN)

****  THE SCALAR PRODUCT OF DN AND DK

      PROD=0
      DO 30 I=1,3
 30   PROD=PROD+DK(I)*DN(I)
```

```
****    NEW DK = OLD DK- 2*PROD*DN

        DO 40 I=1,3
 40     DK(I)=DK(I)-2*PROD*DN(I)

***     TRANSFORM BACK TO THE MIRROR SYSTEM

        CALL TRANSLATE(X,XYTA)
        R2=-R2
        CALL ROTATE(X(3),X(1),R1,R2)
        CALL ROTATE(DK(3),DK(1),R1,R2)
        R2=-R2

***     TEST IF THE RAY HAS HIT THE MIRROR

        DO  I=1,2
        IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
        ENDDO
        IF (S.LT.0) ITRAFF=-ITRAFF

        RETURN
        END
```

```
      SUBROUTINE REFLECT(X,DK)

****  THIS ROUTINE CALCULATES THE REFLECTION OF
****  A RAY GIVEN BY X AND DK WITH THE Z=0 PLANE

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3)

      IF(DK(3).EQ.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=-X(3)/DK(3)
      DO 20 I=1,2
20    X(I)=X(I)+DK(I)*S
      X(3)=0.

      DK(3)=-1.*DK(3)

      ITRAFF=1
      DO I=1,2
      IF ((X(I).LT.XMIN(I)).OR.(X(I).GT.XMAX(I))) ITRAFF=2
      ENDDO
      IF (S.LT.0) ITRAFF=-ITRAFF

      RETURN
      END
```

```
      SUBROUTINE ROTATE(X1,X2,R1,R2)

***   THIS ROUTINE CALCULATES NEW COORDINATES IN A SYSTEM WHICH
***   HAS BEEN ROTATED AN ANGLE ´-A´ AROUND THE X3-AXIS WHERE
***   X1,X2,X3 FORMS A RIGHT-HANDED COORDINATE SYSTEM.
***   R1 AND R2 ARE GIVEN BY:
***   R1=COS(A)
***   R2=SIN(A)
***   OBSERVE THAT IN THE FIGURE BELOW (A.GT.0) GIVING (R1.GT.0)
***   AND (R2.GT.0)
***                                      X2
***                                     I   .X2-NEW
***                                .    I  .
***                                   . I .
***                                  . I.
***                           ------------------   X1
***                                  .I  .A
***                                 . I     .
***                                .  I        .X1-NEW
***                                   I
***
**********************************************************************

      IMPLICIT REAL*8 (A-H,O-Z)

      Y1=R1*X1-R2*X2
      Y2=R2*X1+R1*X2
      X1=Y1
      X2=Y2

      RETURN
      END
```

```
      SUBROUTINE SPHERE(X,DK,R)

***   THIS SUBROUTINE REFLECTS A RAY GIVEN BY X AND DK
***   IN THE SPHERE

***   X**2 + Y**2 + (Z-R)**2 = R**2

***   THE X-DIRECTION COSINE (DK(1)) MUST BE POSITIVE

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3),GRAD(3)

      P=X(1)**2+X(3)**2+X(2)**2-2.*X(3)*R
      Q=X(1)*DK(1)+X(3)*DK(3)+X(2)*DK(2)-DK(3)*R

      IF(Q**2-P.LT.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

***   THE INTERSECTION PARAMETER S IS CALCULATED

      S=-Q+SQRT(Q*Q-P)

***   THE INTERSECTION POINT IS GIVEN BY

      X(1)=X(1)+S*DK(1)
      X(2)=X(2)+S*DK(2)
      X(3)=X(3)+S*DK(3)

      ITRAFF=1
      DO I=1,2
      IF(X(I).LT.XMIN(I).OR.X(I).GT.XMAX(I)) ITRAFF=2
      ENDDO
      IF (S.LT.0) ITRAFF=-ITRAFF

***   THE NORMAL VECTOR

      GRAD(1)=-2.*X(1)
      GRAD(3)=-2.*(X(3)-R)
      GRAD(2)=-2.*X(2)
      CALL NRMLIZ(GRAD)

***   THE MIRROR EQUATION

      PROD=0
      DO I=1,3
      PROD=PROD+GRAD(I)*DK(I)
      ENDDO
```

```
DO I=1,3
DK(I)=DK(I)-2*PROD*GRAD(I)
ENDDO

RETURN
END
```

```
      SUBROUTINE SPHERGRAT(X,DK,PAR,RADIUS)

***   THIS ROUTINE CALCULATES THE DIFFRACTION OF
***   A RAY GIVEN BY X AND DK IN A SPHERICAL GRATING
***   GIVEN BY

***   X**2 + Y**2 + (Z-RADIUS)**2 = RADIUS**2

***   THE X-DIRECTION COSINE (DK(1)) MUST BE POSITIVE.
***   THE GRATING EQUATION IS TAKEN IN A LOCAL SYSTEM
***   RTAK PTAK AND QTAK.
***   THE GRATING IS CHARACTERIZED BY RADIUS AND
***   PAR= WAVELENGTH*GROOVE DENSITY

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.


      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3),PTAK(3),QTAK(3),RTAK(3)

***   THE POINT OF INTERSECTION WITH THE GRATING SURFACE

      Q=X(1)*DK(1)+X(2)*DK(2)+X(3)*DK(3)-RADIUS*DK(3)
      R=DOT(X,X)-2*RADIUS*X(3)

      IF(Q**2-R.LT.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=-Q+SQRT(Q**2-R)

      X(1)=X(1)+DK(1)*S
      X(2)=X(2)+DK(2)*S
      X(3)=X(3)+DK(3)*S

      ITRAFF=1
      DO I=1,2
      IF ((X(I).GT.XMAX(I)).OR.(X(I).LT.XMIN(I))) ITRAFF=2
      ENDDO
      IF(S.LT.0) ITRAFF=-ITRAFF

***   VECTORS OF LOCAL SYSTEM PTAK,QTAK,RTAK
***   THE GRATING EQUATION IS TAKEN IN THIS SYSTEM

***   FIRST RTAK DIRECTED ALONG THE SURFACE NORMAL

      RTAK(1)=-2*X(1)
      RTAK(2)=-2*X(2)
      RTAK(3)=-2*(X(3)-RADIUS)
      CALL NRMLIZ(RTAK)
```

```
***     THEN QTAK ORTHOGONAL TO RTAK AND THE X-AXIS OF THE
***     GRATING SYSTEM.
***     THIS MEANS THAT THE GROOVES ARE PARALLELL CIRCLES

        QTAK(1)=0
        QTAK(2)=1.
        QTAK(3)=-1.*RTAK(2)/RTAK(3)
        CALL NRMLIZ(QTAK)

***     PTAK IS GIVEN BY THE RULES OF A RIGHT HANDED ORTHOGONAL BASIS

        PTAK(1)=QTAK(2)*RTAK(3)-QTAK(3)*RTAK(2)
        PTAK(2)=QTAK(3)*RTAK(1)
        PTAK(3)=-QTAK(2)*RTAK(1)
        CALL NRMLIZ(PTAK)

***     DK IS TURNED AROUND TO POINT OUTWARD FORM GRATING SURFACE

        DO I=1,3
        DK(I)=-DK(I)
        ENDDO

***     THE GRATING EQUATION

        Q=DOT(DK,RTAK)+SQRT(DOT(DK,RTAK)**2-PAR**2-2*PAR*DOT(DK,PTAK))

        DK(1)=-DK(1)-PAR*PTAK(1)+Q*RTAK(1)
        DK(2)=-DK(2)-PAR*PTAK(2)+Q*RTAK(2)
        DK(3)=-DK(3)-PAR*PTAK(3)+Q*RTAK(3)

        RETURN
        END


*************************************************************
      FUNCTION DOT(X,Y)
*************************************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION X(3),Y(3)

        DOT=0.
        DO I=1,3
        DOT=DOT+X(I)*Y(I)
        ENDDO

        RETURN
        END
```

```
      SUBROUTINE SRSOURCE(X,DK,ISEED,HORANG,FWHM,UTSTR,RADIUS)

***   THIS SUBROUTINE SIMULATES THE SYNCHROTRON RADIATION
***   FROM A BENDING MAGNET
***   THE POINT OF ORIGIN, X, AND DIRECTION, DK, OF ONE
***   RAY IS CALCULATED
***   THE SOURCE VOLUME IS A SECTION OF THE ELECTRON BEAM
***   WITH A LENGTH GIVEN BY THE RADIUS OF THE BENDING MAGNET
***   (RADIUS) AND THE TOTAL HORIZONTAL ANGLE OF RADIATION
***   (HORANG). THE CROSS SECTION OF THE ELECTRON BEAM IS
***   CIRCULAR WITH A GAUSSIAN CURRENT DISTRIBUTION HAVING
***   A FULL WIDTH AT HALF MAXIMUM = UTSTR.
***   THE SOURCE POINT (X) IS CHOSEN WITHIN THE SOURCE VOLUME
***   USING A RANDOM DISTRIBUTION SAMPLING THE CURRENT DENSITY.
***   THE DIRECTION OF THE RAY (DK) LIES IN A VERTICAL PLANE
***   TANGENTIAL TO THE CIRCULAR ELECTRON TRAJECTORY. THE ANGLE
***   BETWEEN THE RAY AND THE HORIZONTAL PLANE IS CHOSEN RANDOMLY
***   WITH A GAUSSIAN DISTRIBUTION HAVING A FULL WIDTH AT HALF
***   MAXIMUM = FWHM
***   THE LOCAL COORDINATE SYSTEM HAS ITS ORIGO IN THE CENTER
***   OF THE SOURCE VOLUME. THE X-AXIS IS TANGENTIAL TO THE
***   CENTRAL ELECRON TRAJECTORY AND POINTING IN THE DIRECTION
***   OF THE BEAM. THE Y-AXIS IS POINTING TOWARDS THE CENTER
***   OF THE BENDING MAGNET.


      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(3),DK(3)

      SLUSK=1.
      PI=4*ATAN(SLUSK)

      PSISIG=FWHM/2.35
      RANSIG=UTSTR/2.35

      HOR=HORANG*(RAN(ISEED)-0.5)
      X(1)=-RADIUS*SIN(HOR)
      X(2)=-RADIUS*COS(HOR)+RADIUS

      CALL NORMAL(RANSIG,RAND,ISEED)
      RAND=ABS(RAND)
      GAMMA=2*PI*RAN(ISEED)
      X(1)=X(1)+RAND*COS(GAMMA)*SIN(HOR/2.)
      X(2)=X(2)+RAND*COS(GAMMA)*COS(HOR/2.)
      X(3)=RAND*SIN(GAMMA)

      CALL NORMAL(PSISIG,PSI,ISEED)
      DK(1)=COS(PSI)*COS(HOR)
      DK(2)=-COS(PSI)*SIN(HOR)
      DK(3)=SIN(PSI)

      RETURN
      END
```

```
**********************************************************************
***
      SUBROUTINE NORMAL(SIGMA,X,ISEED)
***
***   THIS SUBROUTINE GENERATES A RANDOM NUMBER, X, WITH A NORMAL
***   DISTRIBUTION. THE STANDARD DEVIATION IS SIGMA AND THE MEAN
***   VALUE IS 0.

      IMPLICIT REAL*8 (A-H,O-Z)

      X=SIGMA*SQRT(-2.*LOG(RAN(ISEED)))
      U=6.283185*RAN(ISEED)
      X=X*SIN(U)

      RETURN
      END
```

```
      SUBROUTINE TORGRAT(X,DK,RMAJ,RMIN,PAR)

***   THIS SUBROUTINE DIFFRACTS A RAY GIVEN BY X AND DK IN
***   A TOROIDAL GRATING WITH MINOR AND MAJOR AXES RMIN RESP RMAJ
***   THE GRATING MAKES A TANGENT TO THE XY PLANE IN ORIGO. THE
***   LARGEST RADIUS OF CURVATURE LIES IN THE XZ PLANE
***   THE TOROID SURFACE IS GIVEN IN FUNCTION  F
***   THE GRADIENT OF THE SURFACE IS GIVEN BY GRADF

***   THE Z-DIRECTION COSINE (DK(3)) MUST BE NEGATIVE.

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3),GRADF(3),RTAK(3),PTAK(3),QTAK(3)

***   THE INTERSECTION POINT BETWEEN THE RAY AND THE SURFACE
***   IS FOUND BY NEWTON RAPHSON ITERATION. STARTING POINT IS
***   THE POINT OF INTERSECTION WITH THE Z=0 PLANE

      IF(DK(3).EQ.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=-X(3)/DK(3)
      XKOLL=X(3)
      X(1)=X(1)+DK(1)*S
      X(2)=X(2)+DK(2)*S
      X(3)=X(3)+DK(3)*S

***   NEWTON RAPHSON ITERATION TO FIND THE INTERSECTION
***   BETWEEN THE RAY AND THE TOROID SURFACE

      IF((RMIN**2-X(2)**2).LT.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

   20 FY=RMIN-SQRT(RMIN**2-X(2)**2)
      F=X(1)**2+(X(3)-RMAJ)**2-(RMAJ-FY)**2

      GRADF(1)=2*X(1)
      GRADF(2)=2*(RMAJ-FY)*X(2)/SQRT(RMIN**2-X(2)**2)
      GRADF(3)=2*(X(3)-RMAJ)

      S=-F/DOT(GRADF,DK)
      X(1)=X(1)+S*DK(1)
      X(2)=X(2)+S*DK(2)
      X(3)=X(3)+S*DK(3)
      IF(ABS(S).GT.1.E-10) GOTO 20
```

```
      FY=RMIN-SQRT(RMIN**2-X(2)**2)

      GRADF(1)=2*X(1)
      GRADF(2)=2*(RMAJ-FY)*X(2)/SQRT(RMIN**2-X(2)**2)
      GRADF(3)=2*(X(3)-RMAJ)

      ITRAFF=1
      DO I=1,2
      IF (X(I).GT.XMAX(I)) ITRAFF=2
      IF (X(I).LT.XMIN(I)) ITRAFF=2
      ENDDO

      SKOLL=(X(3)-XKOLL)/DK(3)
      IF(SKOLL.LT.0) ITRAFF=-ITRAFF

***   NOW CALCULATE VECTORS PTAK,QTAK,RTAK IN LOCAL SYSTEM
***   WHICH IS USED FOR GENERAL GRATING EQUATION

***   FIRST RTAK WHICH IS ORTHOGONAL TO THE GRATING SURFACE
***   OBSERVE THAT GRADF POINTS OUTWARD FROM SURFACE I.E.
***   IN ORIGO GRADF GOES IN NEGATIVE Z DIRECTION

      DO I=1,3
      RTAK(I)=-GRADF(I)
      ENDDO
      CALL NRMLIZ(RTAK)

***   THEN QTAK ORTHOGONAL TO RTAK AND THE X-AXIS IN THE XYZ SYSTEM
***   THIS MEANS THAT THE GROOVES ARE PARALLELL CIRCLES

      QTAK(1)=0
      QTAK(2)=1.
      QTAK(3)=-1.*RTAK(2)/RTAK(3)
      CALL NRMLIZ(QTAK)

***   PTAK IS THEN TAKEN AS THE LAST VECTOR IN A RIGHT HANDED
***   SYSTEM

      PTAK(1)=QTAK(2)*RTAK(3)-QTAK(3)*RTAK(2)
      PTAK(2)=QTAK(3)*RTAK(1)
      PTAK(3)=-QTAK(2)*RTAK(1)
      CALL NRMLIZ(PTAK)

***   THE GENERAL GRATING EQUATION

      Q=-DOT(DK,RTAK)+SQRT(DOT(DK,RTAK)**2-PAR**2+2*PAR*DOT(DK,PTAK))

      DK(1)=DK(1)-PAR*PTAK(1)+Q*RTAK(1)
      DK(2)=DK(2)-PAR*PTAK(2)+Q*RTAK(2)
      DK(3)=DK(3)-PAR*PTAK(3)+Q*RTAK(3)

      RETURN
      END
```

```
**********************************************************************

      FUNCTION DOT(X,Y)

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(3),Y(3)

      DOT=0.
      DO I=1,3
      DOT=DOT+X(I)*Y(I)
      ENDDO
      RETURN
      END
```

```
      SUBROUTINE TOROID(X,DK,RMAJ,RMIN)

***   THIS SUBROUTINE REFLECTS A RAY GIVEN BY X AND DK IN
***   A TOROIDAL MIRROR WITH MINOR AND MAJOR AXES RMIN AND
***   RMAJ, RESP.
***   THE MIRROR MAKES A TANGENT TO THE XY PLANE IN ORIGO.
***   THE LARGEST RADIUS OF CURVATURE LIES IN THE XZ PLANE.
***   THE TOROID SURFACE IS GIVEN IN FUNCTION  F.
***   THE GRADIENT OF THE SURFACE IS GIVEN BY GRADF.

***   THE Z-DIRECTION COSINE (DK(3)) MUST BE NEGATIVE.

***   ITRAFF=0 MEANS NO INTERSECTION WITH SURFACE
***   ITRAFF=1 OR -1 MEANS INTERSECTION WITHIN BOUNDARIES
***   ITRAFF=2 OR -2 MEANS INTERSECTION OUTSIDE BOUNDARIES
***   NEGATIVE ITRAFF MEANS THAT THE RAY HAS ALREADY
***   PASSED THE ELEMENT SURFACE.

      IMPLICIT REAL*8(A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION X(3),DK(3),GRADF(3),RTAK(3),PTAK(3),QTAK(3)

***   THE INTERSECTION POINT BETWEEN THE RAY AND THE SURFACE
***   IS FOUND BY NEWTON RAPHSON ITERATION. STARTING POINT IS
***   THE POINT OF INTERSECTION WITH THE Z=0 PLANE

      IF(DK(3).EQ.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

      S=-X(3)/DK(3)
      XKOLL=X(3)
      X(1)=X(1)+DK(1)*S
      X(2)=X(2)+DK(2)*S
      X(3)=X(3)+DK(3)*S

***   NEWTON RAPHSON ITERATION TO FIND THE INTERSECTION
***   BETWEEN THE RAY AND THE TOROID SURFACE

      IF((RMIN**2-X(2)**2).LT.0) THEN
      ITRAFF=0
      RETURN
      ENDIF

   20 FY=RMIN-SQRT(RMIN**2-X(2)**2)
      F=X(1)**2+(X(3)-RMAJ)**2-(RMAJ-FY)**2

      GRADF(1)=2*X(1)
      GRADF(2)=2*(RMAJ-FY)*X(2)/SQRT(RMIN**2-X(2)**2)
      GRADF(3)=2*(X(3)-RMAJ)

      S=-F/DOT(GRADF,DK)
      X(1)=X(1)+S*DK(1)
      X(2)=X(2)+S*DK(2)
      X(3)=X(3)+S*DK(3)
```

```
      IF(ABS(S).GT.1.E-10) GOTO 20

*****  END OF NEWTON RAPSON

      ITRAFF=1
      DO I=1,2
      IF (X(I).GT.XMAX(I)) ITRAFF=2
      IF (X(I).LT.XMIN(I)) ITRAFF=2
      ENDDO

      SKOLL=(X(3)-XKOLL)/DK(3)
      IF(SKOLL.LT.0) ITRAFF=-ITRAFF

      FY=RMIN-SQRT(RMIN**2-X(2)**2)
      GRADF(1)=2*X(1)
      GRADF(2)=2*(RMAJ-FY)*X(2)/SQRT(RMIN**2-X(2)**2)
      GRADF(3)=2*(X(3)-RMAJ)
      CALL NRMLIZ(GRADF)

      PROD=0
      DO I=1,3
      PROD=PROD+DK(I)*GRADF(I)
      ENDDO

      DO I=1,3
      DK(I)=DK(I)-2*PROD*GRADF(I)
```

APPENDIX 3

A plane grating monochromator - an example of ray tracing.

In order to give the reader a non trivial example of how a ray tracing program looks like we include the FORTRAN code for the MAIN program and the STARTUP subroutine for the case of a Plane Grating Monochromator studied for the MAX synchrotron facility. The local coordinate systems used in the programs and the optical element surfaces are described in figure 8 and figure 9 below. The source is simulated by the subroutine SRSOURCE.

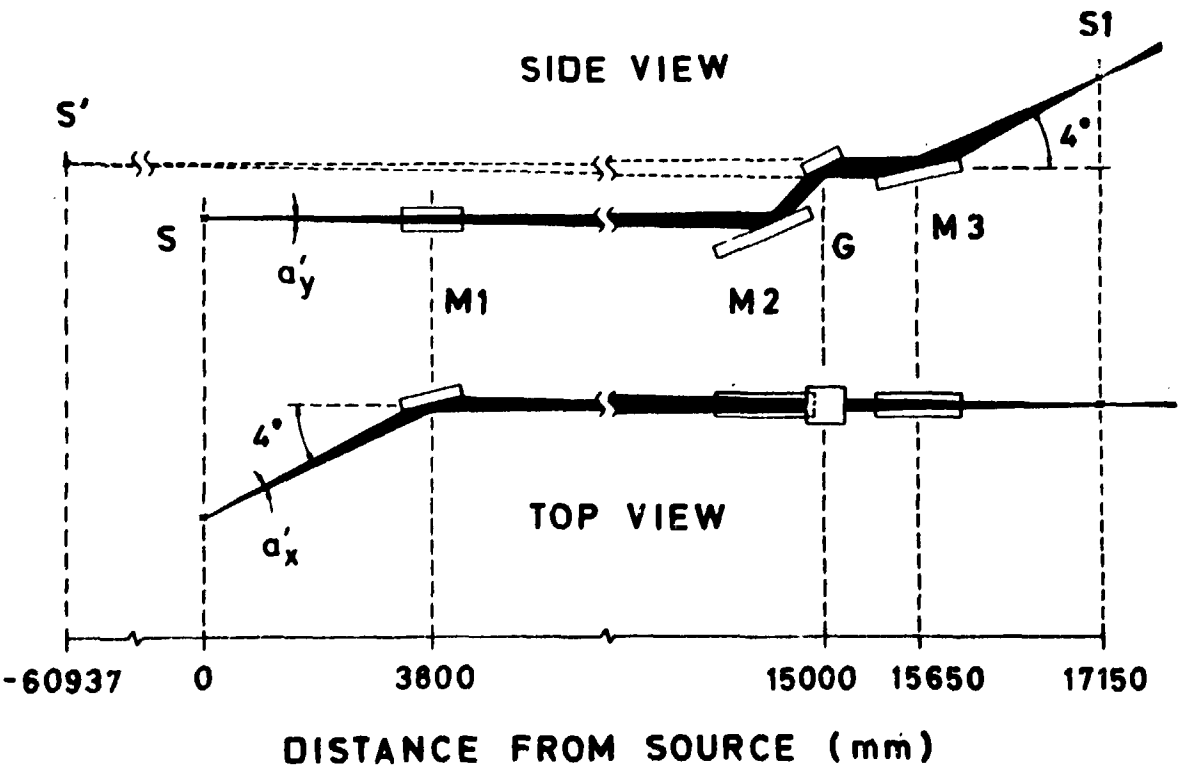
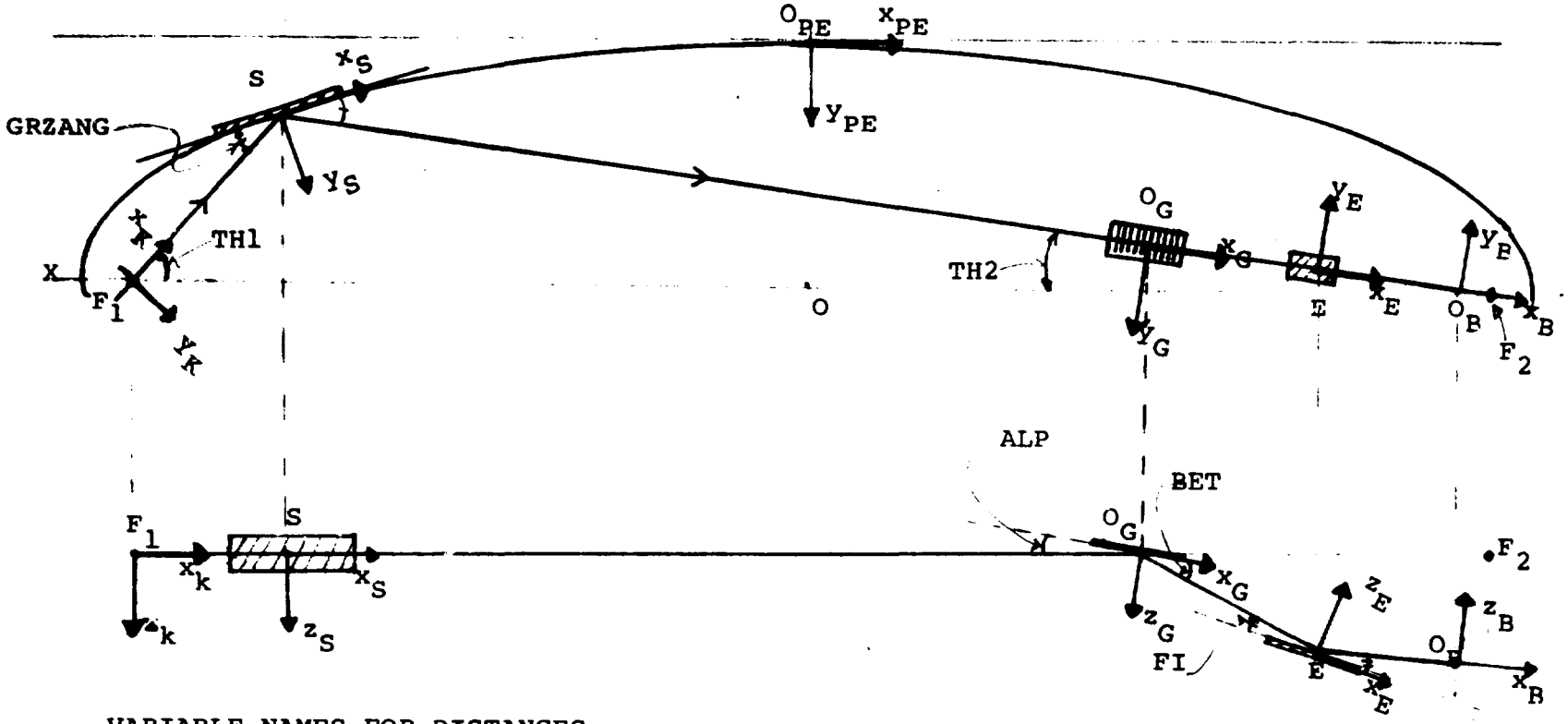


Figure 8. The side and top view of the plane grating (PGM) monochromator.S is the source. M1 is a plane elliptic collecting mirror. M2 is a rotable plane mirror. G is the plane grating. M3 is a plane elliptic focussing mirror. S1 is the exit slit.



VARIABLE NAMES FOR DISTANCES

APL1	=X-O	STILLG	$=F_1-S-O_G$
BPL1	$=O_{PE}-O$	GTILLE	$=O_G-E$
CPL1	$=F_1-O$	RR2	$=E-O_B$
DIST	$=F_1-S$		
RPL2	$=S-F_2=S-O_G-E-O_B$		

FIX FOCUS MODE:

$$\frac{\sin(BET)}{\sin(ALP)} = 2.25$$

Figure 9. The coordinate systems used in program PGM. The plane mirror is omitted from the ray tracing.

```
      PROGRAM PGM

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON XMIN(3),XMAX(3),ITRAFF
      DIMENSION XSOUR(3),XKALL(3),XGITT(3),XELL(3),XIM(3)
      DIMENSION DK(3),R1(4),R2(4),X(3)
      CHARACTER ISTRUNT,YNOUT,YNTX

      SLUSK=1.
      PI=4*ATAN(SLUSK)
     NTOTIN=0
     NTOTUT=0
     NSLIT=0

****  INPUT DISTANCES FOR OPTICAL ELEMENTS

     WRITE(*,*)'ALL DISTANCES SHOULD BE GIVEN IN MM'
     WRITE(*,*)'DISTANCE SOURCE-COLLECTING MIRROR'
     READ(*,*)DIST
     WRITE(*,*)'DISTANCE SOURCE-GRATING'
     READ(*,*)STILLG
     WRITE(*,*)'DISTANCE GRATING-FOCUSSING MIRROR'
     READ(*,*)GTILLE
     WRITE(*,*)'FOCAL DISTANCE FOR FOCUSSING MIRROR'
     READ(*,*)RR2
     WRITE(*,*)'DISTANCE FROM EXIT SLIT TO FOCUS FOR
    C  COLLECTING MIRROR'
     READ(*,*)DRPL2
     WRITE(*,*)'DISTANCE FROM SLIT TO IMAGE PLANE'
     READ(*,*)DSLIT

****  WAVELENGTH (Å) , GROOVE DENSITY (MM-1)
****  AND SIN(BETA)/SIN(ALFA)

     WRITE(*,*)'GIVE LAMBDA FOR SET-UP(Å)'
     READ(*,*)DLAMBDA
     WRITE(*,*)'GROOVES/MM AND SIN(B)/SIN(A)'
     READ(*,*)DNRITS,DC
      PAR=DLAMBDA*DNRITS*1.E-7

****  GRAZING ANGLES FOR MIRRORS

     WRITE(*,*)'GRAZING ANGLES FOR 1:ST AND 2:ND MIRROR (DEGREES)'
     READ(*,*)GRZANG,FI

     GRZ=GRZANG*PI/180.
     FIRAD=FI*PI/180.

****  FOCAL DISTANCE FOR COLLECTING MIRROR

     RPL2=STILLG-DIST+GTILLE+RR2+DRPL2

****  OBJECT DISTANCE FOR FOCUSSING MIRROR

     RR1=STILLG*DC**2+GTILLE
```

```
      NZER=1
      WRITE(*,*)'ZERO ORDER IMAGE? (Y/N)'
      READ(*,111)ISTRUNT
      IF(ISTRUNT.EQ.'Y') THEN
      PAR=0.
      DC=1.
      NZER=0
      ELSE
       WRITE(*,*)' '
       WRITE(*,*)'LAMBDA, GROOVES/MM, SIN(B)/SIN(A)'
       WRITE(*,100)DLAMBDA,DNRITS,DC
  100  FORMAT(1X,3F12.4)
      ENDIF

****   STARTUP CALCULATES TRANSLATION AND ROTATION VECTORS
****   AND FOLLOWS THE PRINCIPAL RAY THROUGH THE SYSTEM

       CALL STARTUP(PAR,DC,STILLG,FI,GTILLE,RR1,RR2,XKALL,
     1 XGITT,XELL,XIM,R1,R2,A,B,DIST,RPL2,GRZANG,APL1,BPL1)
      XSOUR(1)=DIST
      XSOUR(2)=0.
      XSOUR(3)=0.
      XIM(1)=DSLIT
      XIM(2)=0.
      XIM(3)=0.

       WRITE(*,*)' '
       WRITE(*,*)'WHAT ABOUT SOME RAYS ?'
       READ(*,111)ISTRUNT
  111  FORMAT(A1)

      WRITE(*,*)'ISEED'
      READ(*,*)ISEED

****   CHOOSE WAVELENGTH FOR RAYTRACE

      IF (NZER.EQ.1) THEN
      WRITE(*,*)'WAVELENGTH FOR RAYTRACE (A)'
      READ(*,*)DLAMBDA
        PAR=DLAMBDA*DNRITS*1.E-7
      ENDIF

      WRITE(*,*)'LENGTH OF COLLECTING MIRROR'
      READ(*,*)COLLX
      WRITE(*,*)'LENGTH OF GRATING '
      READ(*,*)GITTLENGTH
      WRITE(*,*)'LENGTH OF FOCUSSING MIRROR'
      READ(*,*)FOCX
      WRITE(*,*)'EXIT SLIT IN MICROMETERS'
      READ(*,*)SLIT

      WRITE(*,*)'TEKTRONIX PLOT? (Y/N)'
      READ(*,111)YNTX

      WRITE(*,*)'OUTPUT TO FILE? (Y/N)'
      READ(*,111)YNOUT
```

```
****   DEFINITION OF SOURCE

       WRITE(*,*)'SOURCE HEIGHT IN MM'
       READ(*,*)UTSTR
       WRITE(*,*)'HORISONTAL SPREAD IN MRAD'
       READ(*,*)HORANG
       HORANG=HORANG*1.E-3
       WRITE(*,*)'VERTICAL FWHM OF RADIATION IN MRAD'
       READ(*,*)FWHM
       FWHM=FWHM*1.E-3
       WRITE(*,*)'RADIUS OF BENDING MAGNET IN MM'
       READ(*,*)RADIUS

****     SOME OUTPUT PARAMETERS

       WRITE(*,*)'HOW MANY RAYS THROUGH'
       READ(*,*)IRAYS
       IF(YNTX.EQ.'Y') THEN
       WRITE(*,*)'GIVE PLOT SCALES (XSCALE,YSCALE)'
       READ(*,*)XSCALE,YSCALE
       WRITE(*,*)'READY TEKTRONIX THEN ENTER ONE CHARACTER TO START'
       READ(*,111)ISTRUNT
       CALL TXPLOT(0,0,0,0,0,1)
       DO J=-1,1,2
       X2=J*SLIT*5.E-4
       DO I=0,100
       X1=(I-50)*XSCALE/50.
       CALL TXPLOT(X1,X2,XSCALE,YSCALE,0,0)
       ENDDO
       ENDDO
       ENDIF

****   START OF RAYTRACING LOOP

 4       CONTINUE

****    GENERATE A RAY

       CALL SRSOURCE(X,DK,ISEED,HORANG,FWHM,UTSTR,RADIUS)
       NTOTIN=NTOTIN+1

****    TO COLLECTING MIRROR

        CALL TRANSLATE(X,XSOUR)
        CALL ROTATE(X(1),X(2),R1(1),R2(1))
        CALL ROTATE(DK(1),DK(2),R1(1),R2(1))

****    REFLECTION IN THE COLLECTING MIRROR

       XMAX(1)=COLLX/2.
       XMIN(1)=-COLLX/2.
       XMAX(3)=20.
       XMIN(3)=-20.

        CALL PLANELLZ(X,DK,DIST,RPL2,GRZ,APL1,BPL1)
        IF(ITRAFF.EQ.0) GOTO 6
```

```
****   TO THE GRATING SYSTEM

       CALL TRANSLATE(X,XKALL)
       CALL ROTATE(X(1),X(2),R1(1),R2(1))
       CALL ROTATE(X(3),X(1),R1(2),R2(2))
       CALL ROTATE(DK(1),DK(2),R1(1),R2(1))
       CALL ROTATE(DK(3),DK(1),R1(2),R2(2))

****   GRATING EQUATION

       XMIN(1)=-GITTLENGTH/2.
       XMAX(1)=GITTLENGTH/2.
       XMIN(2)=-15.
       XMAX(2)=15.

       CALL GRATING(X,DK,PAR)
       IF(ITRAFF.EQ.0) GOTO 6

****   TO FOCUSSING MIRROR

       CALL TRANSLATE(X,XGITT)
       CALL ROTATE(X(3),X(1),R1(3),R2(3))
       CALL ROTATE(DK(3),DK(1),R1(3),R2(3))

****   ROTATION 180 DEGREES AROUND X-AXIS

       X(3)=-X(3)
       X(2)=-X(2)
       DK(2)=-DK(2)
       DK(3)=-DK(3)

****   REFLECTION IN THE FOCUSSING MIRROR

      XMIN(1)=-FOCX/2.
      XMAX(1)=FOCX/2.
      XMIN(2)=-15.
      XMAX(2)=15.

       CALL PLANELLY(X,DK,RR1,RR2,FIRAD,A,B)
      IF(ITRAFF.EQ.0) GO TO 6

****   TO THE EXIT SLIT

      CALL ROTATE(X(3),X(1),R1(4),R2(4))
      CALL ROTATE(DK(3),DK(1),R1(4),R2(4))
       CALL TRANSLATE(X,XELL)
       CALL PLANE(X,DK)

****  CHECK IF THE RAY GOES THROUGH THE EXIT SLIT

      IF(ABS(X(3)).LT.SLIT*1.E-3/2.) NSLIT=NSLIT+1
      CALL TRANSLATE(X,XIM)
      CALL PLANE(X,DK)

      IF(YNTX.EQ.'Y') THEN
      CALL TXPLOT(X(2),X(3),XSCALE,YSCALE,0,0)
      ENDIF
```

```
      IF(YNOUT.EQ.'Y') THEN
        WRITE(9,*)X
      ENDIF

       NTOTUT=NTOTUT+1
      IF(NTOTUT.GE.IRAYS) GOTO 5

  6    CONTINUE
      IF(NTOTIN.GE.10000) GOTO 5
      GOTO 4

  5    CONTINUE

      IF (YNTX.EQ.'Y') THEN
      WRITE(*,*)'TO VT100'
      READ(*,111)ISTRUNT
      ENDIF

       WRITE(*,2000)'NUMBER OF RAYS IN',NTOTIN,'  AND OUT',NTOTUT
 2000  FORMAT(1X,A17,I5,A9,I5)
      WRITE(*,*)'NUMBER OF RAYS THROUGH EXIT SLIT',NSLIT

       END


**********************************************************************


       SUBROUTINE STARTUP(PAR,DC,STILLG,FI,GTILLE,RR1,RR2,
     1 XKALL,XGITT,XELL,XIM,R1,R2,A,B,DIST,RPL2,GRZANG,APL1,BPL1)

****   THIS SUBROUTINE FOLLOWS THE PATH OF THE PRINCIPAL RAY
****   THROUGH THE SYSTEM.
****   TRANSLATION VECTORS XKALL,XGITT,XELL AND ROTATION
****   PARAMETERS R1= COS(ANGLE) OCH R2=SIN(ANGLE) ARE RETURNED

       IMPLICIT REAL*8 (A-H,O-Z)
       COMMON XMIN(3),XMAX(3),ITRAFF
       DIMENSION XKALL(3),XGITT(3),XELL(3),R1(4),R2(4),DK(3),X(3)
       DIMENSION XX(3),XIM(3)

       SLUSK=1.
       PI=4.*ATAN(SLUSK)

****   GRAZING ANGLES FOR THE MIRRORS

       GRZ=GRZANG*PI/180.
       FIRAD=FI*PI/180.

****  CALCULATION OF GLANCING ANGLE OF INCIDENCE ALP FOR THE GRATING
****    AND THE EXIT GLANCING ANGLE BET

       ALP=2/180.*PI
       CALL NEWTON(ALP,PAR,DC)
       BET=ASIN(DC*SIN(ALP))
```

```
****   START VALUES OF X AND DK

       DK(1)=COS(GRZ)
       DK(2)=-SIN(GRZ)
       DK(3)=0.

       X(1)=-DIST*COS(GRZ)
       X(2)=DIST*SIN(GRZ)
       X(3)=0

****   REFLECTION IN COLLECTING MIRROR

      APL1=0
       CALL PLANELLZ(X,DK,DIST,RPL2,GRZ,APL1,BPL1)

****   TRANSFORMATION TO THE LOCAL SYSTEM FOR THE GRATING

      XKALL(1)=(STILLG-DIST)*COS(GRZ)
      XKALL(2)=(STILLG-DIST)*SIN(GRZ)
      XKALL(3)=0.
      R1(1)=COS(GRZ)
      R2(1)=-SIN(GRZ)
      R1(2)=COS(ALP)
      R2(2)=SIN(ALP)

       CALL TRANSLATE(X,XKALL)
       CALL ROTATE(X(1),X(2),R1(1),R2(1))
       CALL ROTATE(X(3),X(1),R1(2),R2(2))
       CALL ROTATE(DK(1),DK(2),R1(1),R2(1))
       CALL ROTATE(DK(3),DK(1),R1(2),R2(2))

****   THE GENERAL GRATING EQUATION

       CALL GRATING(X,DK,PAR)

****    TRANSFORMATION TO THE LOCAL SYSTEM FOR FOCUSSING MIRROR

      XGITT(1)=GTILLE*COS(BET)
      XGITT(2)=0.
      XGITT(3)=GTILLE*SIN(BET)

      R1(3)=COS(BET-FIRAD)
      R2(3)=SIN(BET-FIRAD)

       CALL TRANSLATE(X,XGITT)
       CALL ROTATE(X(3),X(1),R1(3),R2(3))
       CALL ROTATE(DK(3),DK(1),R1(3),R2(3))

       X(3)=-X(3)
       X(2)=-X(2)
       DK(3)=-DK(3)
       DK(2)=-DK(2)

****   REFLECTION IN THE FOCUSSING MIRROR
```

```
      A=0
       CALL PLANELLY(X,DK,RR1,RR2,FIRAD,A,B)

****   THE IMAGE POINT IN THE FOCAL PLANE

      R1(4)=COS(FIRAD)
      R2(4)=SIN(FIRAD)
      XELL(1)=RR2
      XELL(2)=0.
      XELL(3)=0.

      CALL ROTATE(X(3),X(1),R1(4),R2(4))
      CALL ROTATE(DK(3),DK(1),R1(4),R2(4))
       CALL TRANSLATE(X,XELL)
       CALL PLANE(X,DK)

****   CHECK-VALUES OF PARAMETERS ARE WRITTEN ON TERMINAL

       WRITE(*,*)'AXES AND GRAZING ANGLE OF COLLECTING MIRROR'
       WRITE(*,100)APL1,BPL1,GRZANG
       ALFA=ALP*180/PI
       BETA=BET*180/PI
       THETA=(ALFA+BETA)/2
       WRITE(*,*)'ANGLES FOR GRATING AND PLANE MIRROR'
       WRITE(*,*)' (ALFA,BETA,THETA)'

       WRITE(*,100)ALFA,BETA,THETA
       WRITE(*,*)'AXES AND GRAZING ANGLE OF FOCUSSING MIRROR'
       WRITE(*,100)A,B,FI

       WRITE(*,*)'****** COORDINATES IN THE SOURCE SYSTEM ******'
       XX(1)=DIST
       XX(2)=0.
       XX(3)=0.
       WRITE(*,110)'COLLECTING MIRROR',XX
       XX(1)=XX(1)+(STILLG-DIST)*COS(GRZ*2.)
       XX(2)=XX(2)+(STILLG-DIST)*SIN(GRZ*2.)
       WRITE(*,110)'GRATING          ',XX
       XX(1)=XX(1)+GTILLE*COS(ALP+BET)*COS(GRZ*2.)
       XX(2)=XX(2)+GTILLE*COS(ALP+BET)*SIN(GRZ*2.)
       XX(3)=GTILLE*SIN(ALP+BET)
       WRITE(*,110)'FOCUSSING MIRROR ',XX
       XX(1)=XX(1)+RR2*COS(ALP+BET-FIRAD*2.)*COS(GRZ*2.)
       XX(2)=XX(2)+RR2*COS(ALP+BET-FIRAD*2.)*SIN(GRZ*2.)
       XX(3)=XX(3)+RR2*SIN(ALP+BET-FIRAD*2.)
       WRITE(*.110)'THE SLIT         ',XX
  100  FORMAT(1X,3F12.4)
  110  FORMAT(1X,A16,3F12.4)
       WRITE(*,*)' '

       WRITE(*,*)'CHECK IF PRINCIPAL RAY HITS ORIGO?',X

       RETURN
       END
```

```
*************************************************************

      SUBROUTINE NEWTON(X,PAR,C)

***   THIS ROUTINE CALCULATES THE ANGLE ALFA AS A SOLUTION TO THE
***   GRATING EQUATION:
****   F(X) = COSX - COS(ARCSIN(C*SINX)) - PAR = 0
****   PAR = GROOVE DENSITY * WAVELENGTH
****   C = SIN(BETA)/SIN(ALFA)

      IMPLICIT REAL*8 (A-H,O-Z)

 10   XO=X
      F=COS(X)-COS(ASIN(C*SIN(X)))-PAR
      F1=SIN(X)+SIN(ASIN(C*SIN(X)))/SQRT(1-(C*SIN(X))**2)*C*COS(X)
      X=XO-F/F1
      IF (ABS(X-XO).GT.1.E-10) GO TO 10

      RETURN
      END


*************************************************************
*                                                           *
*  THIS SUBROUTINE GIVES A PLOT ON A TEKTRONIX 4006-1       *
*  The point to be plotted should be within +-XSCALE        *
*  and +-YSCALE.                                            *
*  IERASE=1 causes erase of screen                          *
*  IAXES=1 axes are plotted                                 *
*************************************************************

      SUBROUTINE TXPLOT(X1,X2,XSCALE,YSCALE,IERASE,IAXES)
      DOUBLE PRECISION X1,X2,XSCALE,YSCALE
      BYTE OYH,OYL,OXH,OXL,OB12,OB27,OB29,OB31,OB00,OSL,OSH
      INTEGER OXA,OYA
      OB00=0
      OB12=12
      OB27=27
      OB29=29
      OB31=31

      IF(IERASE.EQ.1) GOTO 997
      IF(IAXES.EQ.1) GOTO 996

*****  PLOT SEQUENCE

      OXA=INT(300.*X1/XSCALE+500.)
      OYA=INT(300.*X2/YSCALE+400.)
      IF(OXA.LT.0 .OR. OXA.GT.1023) GOTO 999
      IF(OYA.LT.0 .OR. OYA.GT.780) GOTO 999
      OYH=32+INT(OYA/32)
      OYL=96+INT(OYA)-32*INT(OYA/32)
      OXH=32+INT(OXA/32)
      OXL=64+INT(OXA)-32*INT(OXA/32)
      WRITE(*,998)OB29,OB00,OYH,OYL,OXH,OXL,
     C  OB00,OYH,OYL,OXH,OXL,OB00
  998   FORMAT(1X,12A1)
```

```
      RETURN

****   ERASE SEQUENCE

  997   WRITE(*,998)0B00,0B00,0B12,0B27,0B00,0B00,
     C  0B00,0B00,0B00,0B00,0B00,0B00
      RETURN

****   PLOT OF AXES (X=0 AND Y=0)

  996   0YH=44
      0YL=112
      0XH=38
      0XL=72
      0SH=57
      0SL=64
      WRITE(*,995)0B29,0B00,0YH,0YL,0XH,0XL,0B00,
     C  0B00,0YH,0YL,0SH,0SL,0B00,0B00,0B00
  995   FORMAT(1X,15A1)
      0YH=35
      0YL=100
      0XH=47
      0XL=84
      0SH=53
      0SL=124
      WRITE(*,995)0B29,0B00,0YH,0YL,0XH,0XL,0B00,
     C  0B00,0SH,0SL,0XH,0XL,0B00,0B00,0B00

  999 RETURN
      END
```