NASA TECHNICAL MEMORANDUM

NASA TM X-1838

NASA TM X-1838

CASE FILE COPY

HELICAL DUCT GEOMETRY ROUTINE FOR THE SHIELDING COMPUTER PROGRAM "FASTER"

by Thomas M. Jordan and Millard L. Wohl

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1969

NASA TM X-1838

HELICAL DUCT GEOMETRY ROUTINE FOR THE SHIELDING COMPUTER PROGRAM "FASTER"

By Thomas M. Jordan and Millard L. Wohl

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

ABSTRACT

A method for tracking neutrons through a slab penetrated by a helical voided duct is presented and verified. FORTRAN listings of computer subroutines that perform the numerical analysis are included; these subroutines are designed to be used with FASTER, a high-efficiency Monte Carlo code. Neutron number spectra were calculated at several points in and near the emergent end of a duct through a 20- and a 100-centimeter thick water slab having a point source at the center of the inlet.

HELICAL DUCT GEOMETRY ROUTINE FOR THE SHIELDING COMPUTER PROGRAM "FASTER"

by Thomas M. Jordan* and Millard L. Wohl

Lewis Research Center

SUMMARY

A method for tracking neutrons through slabs penetrated by helical voided ducts is presented and verified by means of the FASTER computer code. A point fission source is located at the center of the entrant mouth of the duct. Neutron number spectra computed at point detectors in and near the emergent mouth of the duct are compared with similar results computed by the FASTER code and the O5R code using cylindrical geometry to describe the duct. The helical duct is given a very large pitch to simulate a cylindrical duct within the slabs considered. The comparison of number spectra is almost exact, verifying the techniques developed herein. FORTRAN listings of computer subroutines that perform the numerical analysis are included; these subroutines are designed to be used in conjunction with FASTER, a high-efficiency Monte Carlo code.

INTRODUCTION

A useful reactor coolant flow passage geometry is that of a helicoid, or a helical tubular duct. This duct configuration has the dual advantage of reducing radiation streaming through a shield and permitting low coolant pressure drop.

In order to track neutrons and gamma rays through such a complex geometric configuration, subroutines are written to be used with the FASTER code. Sample problems (ref. 1), previously run with cylindrical ducts with the FASTER and O5R codes, were solved using a helical duct; this was done by assigning a pitch of 10^5 centimeters to a helicoid penetrating a 20- and a 100-centimeter-thick water slab. Thus, the important within-slab region of the helicoid is essentially cylindrical.

Agreement was obtained for the neutron number spectrum at several point detectors in and near the emergent mouth of a duct having a point fission neutron source at the

*ART Research Corporation, Los Angeles, California.

center of one end of the duct. This agreement verified the helical duct geometry handling subroutines.

ANALYSIS

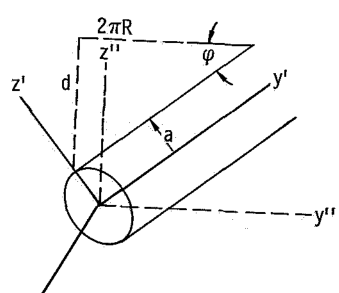
The equation of the helicoid is developed by using the fact that the intersection of the surface and a cutting plane normal to the centerline of the helicoid is circular. The parametric equations of the duct centerline are

$$x = R \cos \omega z \tag{1a}$$

$$y = R \sin \omega z \tag{1b}$$

where $\omega = 2\pi/d$, d is the pitch of the helical centerline, and R is the perpendicular distance from the axis of the cylinder around which the helicoid is wound to the helical centerline.

A point (the origin of the double primed coordinate system is sketch (a)) is first fixed at an arbitrary azimuthal angle θ. Solving for the equivalent z-coordinate gives $z_o = \theta/\omega$. A coordinate system is centered at this point so that its y'-axis is tangent to the helical centerline and its x'-axis is along the cylindrical radius vector as in sketch (a):



(a)

In the primed coordinate system, the intersection of the unwound helical duct with the cutting plane has the following equation:

$$x'^2 + z'^2 = a^2 \tag{2}$$

where a is the inner radius of the duct. Performing a rotation φ about the x'-axis, the appropriate transformation equations become

$$x' = x'' \tag{3a}$$

$$y' = y'' \cos \varphi + z'' \sin \varphi \tag{3b}$$

$$z' = -y'' \sin \varphi + z'' \cos \varphi \tag{3c}$$

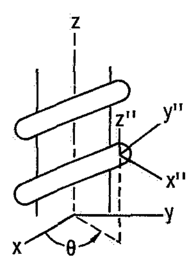
$$\sin \varphi = \frac{d}{\left[d^2 + (2\pi R)^2\right]^{1/2}} = \frac{1}{\left(1 + \omega^2 R^2\right)^{1/2}} \tag{3d}$$

$$\cos \varphi = \frac{2\pi R}{\left[d^2 + (2\pi R)^2\right]^{1/2}} = \frac{\omega R}{\left(1 + \omega^2 R^2\right)^{1/2}} \tag{3e}$$

or

$$x''^2 + (-y'' \sin \varphi + z'' \cos \varphi)^2 = a^2 \tag{4}$$

The double primed coordinate system must also be translated and rotated as indicated in sketch (b) and equations (5).



(b)

$$x'' = x \cos \theta + y \sin \theta - R \tag{5a}$$

$$y'' + -x \sin \theta + y \cos \theta \tag{5b}$$

$$z'' = z - z_o \tag{5c}$$

so that

$$\left(x \cos \theta + y \sin \theta - R\right)^2 + \left[(-x \sin \theta + y \cos \theta) \sin \varphi + (z - z_o) \cos \varphi\right]^2 = a^2 \tag{6}$$

To put everything in terms of the x, y, and z coordinates, the following relations are used:

$$\left.\begin{aligned} \theta &= \tan^{-1}\left(\frac{y}{x}\right) & \sin\theta &= \frac{y}{\left(x^2+y^2\right)^{1/2}} \\ z_o &= \frac{1}{\omega}\tan^{-1}\left(\frac{y}{x}\right) & \cos\theta &= \frac{x}{\left(x^2+y^2\right)^{1/2}} \end{aligned}\right\} \tag{7}$$

The resulting equation is

$$\left[x\frac{x}{\left(x^2+y^2\right)^{1/2}} + y\frac{y}{\left(x^2+y^2\right)^{1/2}} - R\right]^2 + \left\{\left[\frac{-xy}{\left(x^2+y^2\right)^{1/2}} + \frac{xy}{\left(x^2+y^2\right)^{1/2}}\right]\sin\varphi + \left[z - \frac{1}{\omega}\tan^{-1}\left(\frac{y}{x}\right)\right]\frac{\omega R}{\left(1+\omega^2R^2\right)^{1/2}}\right\}^2 = a^2 \tag{8a}$$

or

$$\left[\left(x^2+y^2\right)^{1/2} - R\right]^2 + \frac{R^2}{1+\omega^2R^2}\left[\tan^{-1}\left(\frac{y}{x}\right) - \omega z\right]^2 = a^2 \tag{8b}$$

It should be noted that the arctangent function subroutine will return only the principal value. Therefore, in the actual computations the quantity $\left[\tan^{-1}(y/x) - \omega z\right]$ will be treated as modulo π.

Equation (8b) is generalized to the following form:

$$\varphi(\vec{r}) = \left\{\left[\left(x_j - x_j^o\right)^2 + \left(x_k - x_k^o\right)^2\right]^{1/2} - R\right\}^2 + \frac{R^2}{1+\omega^2R^2}\left[\tan^{-1}\left(\frac{x_k - x_k^o}{x_j - x_j^o}\right) - \omega\left(x_i - x_i^o\right)\right]^2 - a^2 \tag{9}$$

where $\varphi(\vec{r})$ is the helicoid surface function and has the following values:

$$\varphi(\vec{r}) = 0 \quad \text{on the helicoid surface}$$
$$> 0 \quad \text{outside the surface}$$
$$< 0 \quad \text{within the surface}$$

and where $(i,j,k) = (1,2,3), (2,3,1)$, or $(3,1,2)$; $x_1, x_2, x_3 = x, y, z$; and x_i^o, x_j^o, x_k^o are coordinates of the origin of a translated coordinate system.

Thus, the following data are normally supplied to describe the helical duct: $i, x_i^o, x_j^o, x_k^o, d, R, a$.

VALUES OF THE SURFACE FUNCTION

The computer subroutine has been written to evaluate the surface function at the point $\vec{r} = \vec{r} + s\vec{\Omega}$ where $\vec{r}_o$, $\vec{\Omega}$, and S are arguments supplied to the subroutine. Let

$$r_o = (x^*, y^*, z^*) = (x_1^*, x_2^*, x_3^*)$$

$$\vec{\Omega} = (\alpha, \beta, \gamma) = (C_1, C_2, C_3)$$

Then

$$r = (x, y, z) = (x_1, x_2, x_3) = (x_1^* + C_1 S, x_2^* + C_2 S, x_3^* + C_3 S) \tag{10}$$

The value $\varphi(\vec{r})$ of the surface function is obtained by substituting $\vec{r}$ from equation (10) into equation (9).

NORMAL VECTOR

The components of the normal vector n are

$$\vec{n} = (n_1, n_2, n_3) \tag{11}$$

and are calculated as

$$n_l = \frac{\dfrac{\partial\varphi}{\partial x_l}}{\displaystyle\sum_{m=1}^{3}\left(\frac{\partial\varphi}{\partial x_m}\right)^2} \tag{12}$$

and

$$l = 1, 2, 3$$

The partial derivatives are calculated as

$$\frac{\partial\varphi}{\partial x_i} = -2\omega hv \tag{13a}$$

$$\frac{\partial\varphi}{\partial x_j} = 2\left(\frac{\hat{x}_j u}{\zeta} - \frac{hv\hat{x}_k}{\zeta^2}\right) \tag{13b}$$

$$\frac{\partial\varphi}{\partial x_k} = 2\left(\frac{\hat{x}_k u}{\zeta} + \frac{hv\hat{x}_j}{\zeta^2}\right) \tag{13c}$$

where $\left(\hat{x}_i, \hat{x}_j, \hat{x}_k\right) = \left[\left(x_i - x_i^o\right), \left(x_j - x_j^o\right), \left(x_k - x_k^o\right)\right]$ and

$$\zeta = \left[\left(x_j - x_j^o\right)^2 + \left(x_k - x_k^o\right)^2\right]^{1/2} \tag{14a}$$

$$u = \zeta - R \tag{14b}$$

$$v = \tan^{-1}\left(\frac{x_k - x_k^o}{x_j - x_j^o}\right) - \omega\left(x_i - x_i^o\right) \tag{14c}$$

$$h = \frac{R^2}{1 + \omega^2 R^2} \tag{14d}$$

Intersections With Straight Line

The intersection of a straight line with the helicoid is calculated by an iterative process. The assumption is made that the surface function is locally quadratic. Then the value of the function can be expressed by a truncated Taylor series as

$$\varphi(\vec{r}') = \varphi(\vec{r} + S\Omega) = \varphi(\vec{r}) + S\,\frac{\partial\varphi(\vec{r})}{\partial S} + \frac{S^2}{2}\,\frac{\partial^2\varphi(\vec{r})}{\partial S^2} \tag{15}$$

where $\varphi(\vec{r})$ is determined from equation (9), and the coefficient $\partial\varphi(\vec{r})/\partial S$ is calculated from

$$\frac{\partial\varphi(\vec{r})}{\partial S} = \sum_{m=1}^{3} C_m\,\frac{\partial\varphi(\vec{r})}{\partial x_m} \tag{16}$$

using equation (13) to compute the partial derivatives.

The final coefficient is given by

$$\frac{\partial^2\varphi(\vec{r})}{\partial S^2} = \sum_{l=1}^{3}\sum_{m=1}^{3} C_l C_m\,\frac{\partial^2\varphi(\vec{r})}{\partial x_l\,\partial x_m} \tag{17}$$

where

$$\frac{\partial^2\varphi(\vec{r})}{\partial x_i^2} = 2h\,\omega^2 \tag{18a}$$

$$\frac{\partial^2\varphi(\vec{r})}{\partial x_j^2} = 2\left[\frac{u\hat{x}_k^2}{\zeta^3} + \frac{\hat{x}_j^2}{\zeta^2} + h\left(\frac{2v\hat{x}_j\hat{x}_k}{\zeta^4} + \frac{\hat{x}_k^2}{\zeta^4}\right)\right] \tag{18b}$$

$$\frac{\partial^2\varphi(\vec{r})}{\partial x_k^2} = 2\left[\frac{u\hat{x}_j^2}{\zeta^3} + \frac{\hat{x}_k^2}{\zeta^2} + h\left(\frac{-2v\hat{x}_j\hat{x}_k}{\zeta^4} + \frac{\hat{x}_j^2}{\zeta^4}\right)\right] \tag{18c}$$

$$\frac{\partial^2 \varphi(\vec{r})}{\partial x_i \partial x_j} = \frac{2h\omega x_k}{\zeta^2} \tag{18d}$$

$$\frac{\partial^2 \varphi(\vec{r})}{\partial x_j \partial x_k} = 2\left\{\frac{-u\hat{x}_j\hat{x}_k}{\zeta^3} + \frac{\hat{x}_j\hat{x}_k}{\zeta^2} + h\left[\frac{v\left(\hat{x}_k^2 - \hat{x}_j^2\right)}{\zeta^4} - \frac{\hat{x}_j\hat{x}_k}{\zeta^4}\right]\right\} \tag{18e}$$

$$\frac{\partial^2 \varphi(\vec{r})}{\partial x_k \partial x_i} = \frac{-2h\omega\hat{x}_j}{\zeta^2} \tag{18f}$$

The net result is a quadratic equation for the distance S to the surface since the surface function $\varphi(\vec{r})$ must be zero at the intersection

$$\varphi(\vec{r}') = \varphi(\vec{r}) + S\,\frac{\partial\varphi(\vec{r})}{\partial S} + \frac{S^2}{2}\,\frac{\partial^2\varphi(\vec{r})}{\partial S^2} = 0 \tag{19}$$

Let

$$A = \frac{\partial^2\varphi(\vec{r})}{\partial S^2} \tag{20a}$$

$$B = \frac{\partial\varphi(\vec{r})}{\partial S} \tag{20b}$$

$$C = 2\varphi(\vec{r}) \tag{20c}$$

Then

$$AS^2 + 2BS + C + 0 \tag{21a}$$

or

$$S = \frac{-B + \delta\sqrt{B^2 - AC}}{A} \tag{21b}$$

The sign of the square root is determined by the ambiguity index δ (of eq. (21b)) with respect to the region in which the ray tracing is performed. The ambiguity index is defined as

$$\delta = \frac{-\varphi(\vec{r}_g)}{\varphi(\vec{r}_g)} \tag{22}$$

where $\vec{r}_g$ is any point in the region.

Equation (21) is not used when $|AC/B^2| < 10^{-4}$. A one-term expansion is then used for the square root so that

$$S = -\frac{B}{A}\left(1 - \delta \frac{B}{|B|}\right) = \frac{1}{2} \frac{C\delta}{|B|} \tag{23}$$

This procedure eliminated convergence problems caused by underflow in the vicinity of the intersection.

The distance S to the intersection is used to compute a new reference point $\vec{r}$ and the entire procedure is then repeated until the change in S is less than 10^{-5} of the value already obtained. In general, only two iterations are needed.

The changes required to incorporate the helical duct in FASTER are shown in the appendix. The input instructions are modified to include the following on card 2-2:

$i, 0, 16, x^o, y^o, z^o, L, R, a$	i^{th} surface has helix axis parallel to x-axis
$i, 0, 17, y^o, z^o, x^o, L, R, a$	i^{th} surface has helix axis parallel to y-axis
$i, 0, 18, z^o, x^o, y^o, L, R, a$	i^{th} surface has helix axis parallel to z-axis

RESULTS AND DISCUSSION

The duct analysis routine was verified by using it for the 20- and 100-centimeter-thick water slab cylindrical duct problem previously run as shown in figure 1 (ref. 1). Neutron number spectra essentially coincided with previously calculated spectra (O5R and FASTER with cylindrical duct geometry) as seen in figure 2. The cylindrical duct was simulated with the helicoid geometry subroutines by making L, the pitch, equal to 10^5 centimeters. Neutron number spectra at each detector point were calculated using 1000 histories. Computer time was 20 minutes per detector point, approximately double that required for solution of the same problem using the cylindrical geometry routine, as expected because of the more complex geometric description of the cylindrical duct.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, June 2, 1969,
126-15-01-03-22.

APPENDIX - LISTING

Changes to FASTER for Subroutine HELIX

```
C TO INCORPORATE SUBROUTINE HELIX IN FASTER, MAKE THE FOLLOWING CHANGES
C
C SUBROUTINE LOCDUM
C
C INSERT AFTER FAST0151
      IF(MAX.GT.9) CALL HELIX(1,MAX-9,A(1,K),X,X,0.0,U(K))              FAST0151
      IF(MAX.GT.9) GO TO 160                                            FAST0151
C
C SUBROUTINE GEOMIN
C
C REPLACE FAST0532 BY
      GO TO(232,233,235,260,255,282),NGT                                FAST0532
C INSERT AFTER FAST0579
      GO TO 290                                                         FAST0579
  282 DO 284 J=1,6                                                      FAST0579
  284 A(J,I) = AA(J)                                                    FAST0579
      NTP(I) = NEX + 9                                                  FAST0579
C INSERT AFTER FAST0616
      IF(MAX.GT.9) CALL HELIX(1,MAX-9,A(1,K),ADM,ADM,0.0,FST)           FAST0616
      IF(MAX.GT.9) GO TO 335                                            FAST0616
C INSERT AFTER FAST0619
  335 CONTINUE                                                          FAST0619
C
C SUBROUTINE TRADUM
C
C INSERT AFTER FAST2228
      IF(NTP(K).LE.9) GO TO 198                                         FAST2228
      IF(M.GT.1) GO TO 192                                              FAST2228
      SI(1,K) = STT                                                     FAST2228
      SI(2,K) = STT + SP                                                FAST2228
      CALL HELIX(3,KK*(NTP(K)-9),A(1,K),X,C,SI(1,K),SB)                 FAST2228
      IF(SB) 340,330,330                                                FAST2228
  192 CALL HELIX(1,NTP(K)-9,A(1,K),X,C,STT,FST)                         FAST2228
      IF(FLOAT(KK)*FST) 340,340,390                                     FAST2228
  198 CONTINUE                                                          FAST2228
C
C SUBROUTINE NORDUM
C
C INSERT AFTER FAST2343
      IF(MAX.GT.9) CALL HELIX(2,MAX-9,A(1,I),X,X,0.0,C)                 FAST2343
      IF(MAX.GT.9) GO TO 35                                             FAST2343
C INSERT AFTER FAST2355
   35 CONTINUE                                                          FAST2355
```

Subroutine HELIX Listing

```
C***********************************************************************HELIX000
C /JARG/ = 1, HELIX PARALLEL TO X-AXIS  (I,J,K)=(1,2,3)=(X,Y,Z)          HELIX000
C /JARG/ = 2, HELIX PARALLEL TO Y-AXIS  (I,J,K)=(2,3,1)=(Y,Z,X)          HELIX000
C /JARG/ = 3, HELIX PARALLEL TO Z-AXIS  (I,J,K)=(3,1,2)=(Z,X,Y)          HELIX000
C SIGN(JARG) = AMBIGUITY INDEX                                           HELIX000
C VARG(1) = REFERENCE VALUE OF X(I) AT WHICH THETA = 0.0                 HELIX000
C VARG(2) = TRANSLATION OF HELIX AXIS IN X(J) DIRECTION                  HELIX000
C VARG(3) = TRANSLATION OF HELIX AXIS IN X(K) DIRECTION                  HELIX000
C VARG(4) = PERIOD OF HELIX ALONG X(I) AXIS                              HELIX000
C VARG(5) = RADIUS OF GYRATION OF HELIX PERPENDICULAR TO X(I) AXIS       HELIX000
C VARG(6) = RADIUS OF HELIX CROSS SECTION                                HELIX000
C WARG(1) = X COORDINATE OF ORIGIN OF RAY                                HELIX000
C WARG(2) = Y COORDINATE OF ORIGIN OF RAY                                HELIX000
C WARG(3) = Z COORDINATE OF ORIGIN OF RAY                                HELIX000
C XARG(1) = DIRECTION COSINE OF RAY WITH RESPECT TO X AXIS               HELIX000
C XARG(2) = DIRECTION COSINE OF RAY WITH RESPECT TO Y AXIS               HELIX000
C XARG(3) = DIRECTION COSINE OF RAY WITH RESPECT TO Z AXIS               HELIX000
C YARG(1) = DISTANCE ALREADY TRAVERSED ALONG RAY                         HELIX000
C YARG(2) = MAXIMUM DISTANCE ALONG RAY                                   HELIX000
C IARG = 1, COMPUTE VALUE OF HELIX      RETURNED IN ZARG(1)              HELIX000
C IARG = 2, COMPUTE NORMAL VECTOR       RETURNED IN ZARG(1),...,ZARG(3)  HELIX000
C IARG = 3, COMPUTE DISTANCE TO HELIX   RETURNED IN ZARG(1)              HELIX000
C***********************************************************************HELIX000
      SUBROUTINE HELIX(IARG,JARG,VARG,WARG,XARG,YARG,ZARG)               HELIX010
      DIMENSION VARG(1),WARG(1),XARG(1),YARG(1),ZARG(1)                  HELIX020
      DIMENSION XZ(3)   ,X(3)    ,C(3)    ,CN(3)                         HELIX030
      DATA      KOUNT,PI,TPI/0,3.1415927,6.2831853/                      HELIX040
      DO 100 I=1,3                                                       HELIX050
      C(I)     = XARG(I)                                                 HELIX060
      XZ(I)    = WARG(I) + C(I)*YARG(1)                                  HELIX070
  100 X(I)     = XZ(I)                                                   HELIX080
      NGT      = IARG                                                    HELIX090
      I        = IABS(JARG)                                              HELIX100
      J        = I + 1 - 3*(I/3)                                         HELIX110
      K        = J + 1 - 3*(J/3)                                         HELIX120
      TPL      = TPI/VARG(4)                                             HELIX130
      GAM      = 1.0/(TPL**2 + 1.0/VARG(5)**2)                           HELIX140
  110 DXI      = X(I) - VARG(1)                                          HELIX150
      DXJ      = X(J) - VARG(2)                                          HELIX160
      DXK      = X(K) - VARG(3)                                          HELIX170
      USM      = SQRT(DXJ**2 + DXK**2)                                   HELIX180
      THE      = 0.0                                                     HELIX185
      IF(USM.GT.0.0) THE = ATAN2(DXK,DXJ) - TPL*DXI                      HELIX190
      VSM      = THE - TPI*FLOAT(IFIX((THE + PI)/TPI))                   HELIX200
      UBG      = USM - VARG(5)                                           HELIX210
      PHI      = UBG**2 + GAM*VSM**2 - VARG(6)**2                        HELIX220
      IF(NGT.GT.1) GO TO 120                                             HELIX230
      ZARG(1) = PHI                                                      HELIX240
      GO TO 990                                                          HELIX250
  120 CST      = DXJ/USM                                                 HELIX260
      SNT      = DXK/USM                                                 HELIX270
      CN(I)    = -TPL*GAM*VSM                                            HELIX280
      CN(J)    = UBG*CST - GAM*VSM*SNT/USM                               HELIX290
      CN(K)    = UBG*SNT + GAM*VSM*CST/USM                               HELIX300
      DPH      = 0.0                                                     HELIX310
      IF(NGT.GT.2) GO TO 150                                             HELIX320
      DO 130 I=1,3                                                       HELIX330
  130 DPH      = DPH + CN(I)**2                                          HELIX340
      DPH      = SQRT(DPH)                                               HELIX350
      DO 140 I=1,3                                                       HELIX360
  140 ZARG(I) = CN(I)/DPH                                                HELIX370
      GO TO 990                                                          HELIX380
```

```
  150 DO 160 L=1,3
  160 DPH      = DPH + C(L)*CN(L)
      CN(I)    = GAM*TPL**2
      CN(J)    = UBG*SNT**2/USM+CST**2+GAM*SNT*(2.0*VSM*CST+SNT)/USM**2
      CN(K)    = UBG*CST**2/USM+SNT**2-GAM*CST*(2.0*VSM*SNT-CST)/USM**2
      DDP      = 0.0
      DO 170 L=1,3
  170 DDP      = DDP + CN(L)*C(L)**2
      CN(I)    = TPL*GAM*SNT/USM
      CN(J)    = SNT*CST*(1.0 - UBG/USM)
     1         + GAM*(VSM*(SNT**2 - CST**2) - SNT*CST)/USM**2
      CN(K)    = -TPL*GAM*CST/USM
      DO 180 L=1,3
      M        = L + 1 - 3*(L/3)
  180 DDP      = DDP + 2.0*CN(L)*C(L)*C(M)
      IF(NGT.GT.3) GO TO 190
      TOT      = 0.0
      KK       = I/JARG
      SGN      = FLOAT(KK)
      STB      = 2.0*YARG(2)
  190 H        = 0.0
      NEH      = 0
      DLS      = 0.0
      IF(PHI.NE.0.0) GO TO 200
      IF(DDP) 210,250,210
  200 IF(DDP.NE.0.0) GO TO 210
      DLS      = -0.5*PHI/DPH
      GO TO 250
  210 IF(DPH.EQ.0.0) GO TO 220
      IF(ABS(DDP*PHI/DPH**2).GT.1.0E-4) GO TO 220
      ADP      = ABS(DPH)
      DLS      = ADP*(SGN - DPH/ADP)/DDP - 0.5*SGN*PHI/ADP
      GO TO 250
  220 DLS      = -DPH/DDP
      H        = DPH**2 - DDP*PHI
      IF(H) 230,250,240
  230 NEH      = 1
      GO TO 250
  240 DLS      = DLS + SGN*SQRT(H)/DDP
  250 IF((DLS + TOT).LE.0.0) GO TO 290
      TOT      = TOT + DLS
      IF((NEH.GT.0).AND.(NGT.GT.3)) GO TO 290
      IF(NEH.GT.0) GO TO 260
      IF(ABS(DLS/TOT).LE.1.0E-5) GO TO 280
      IF(NGT.GT.4) GO TO 280
  260 DO 270 L=1,3
  270 X(L)     = XZ(L) + TOT*C(L)
      NGT      = NGT + 1
      GO TO 110
  280 STB      = TOT
      IF(DLS.LE.0.0) STB = STB + AMAX1(-DLS,1.0E-5*TOT)
  290 ZARG(1) = STB
      KOUNT    = KOUNT + 1
      IF(KOUNT.LE.100) WRITE(6,2000)NGT,I,J,K,KK,NEH,X,C,DDP,DPH,PHI,
     1TOT,DLS,STB
  990 RETURN
 2000 FORMAT(1X,6I6,1P6E12.4/37X,6E12.4)
      END
```

REFERENCE

1. Wohl, Millard L.; and Jordan, Thomas M.: Use of "FASTER" Code for Significantly Reducing Ducted Reactor Shield Computation Time. NASA TM X-1828, 1969.

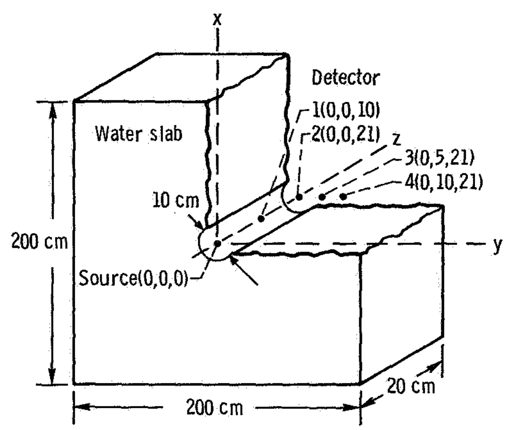
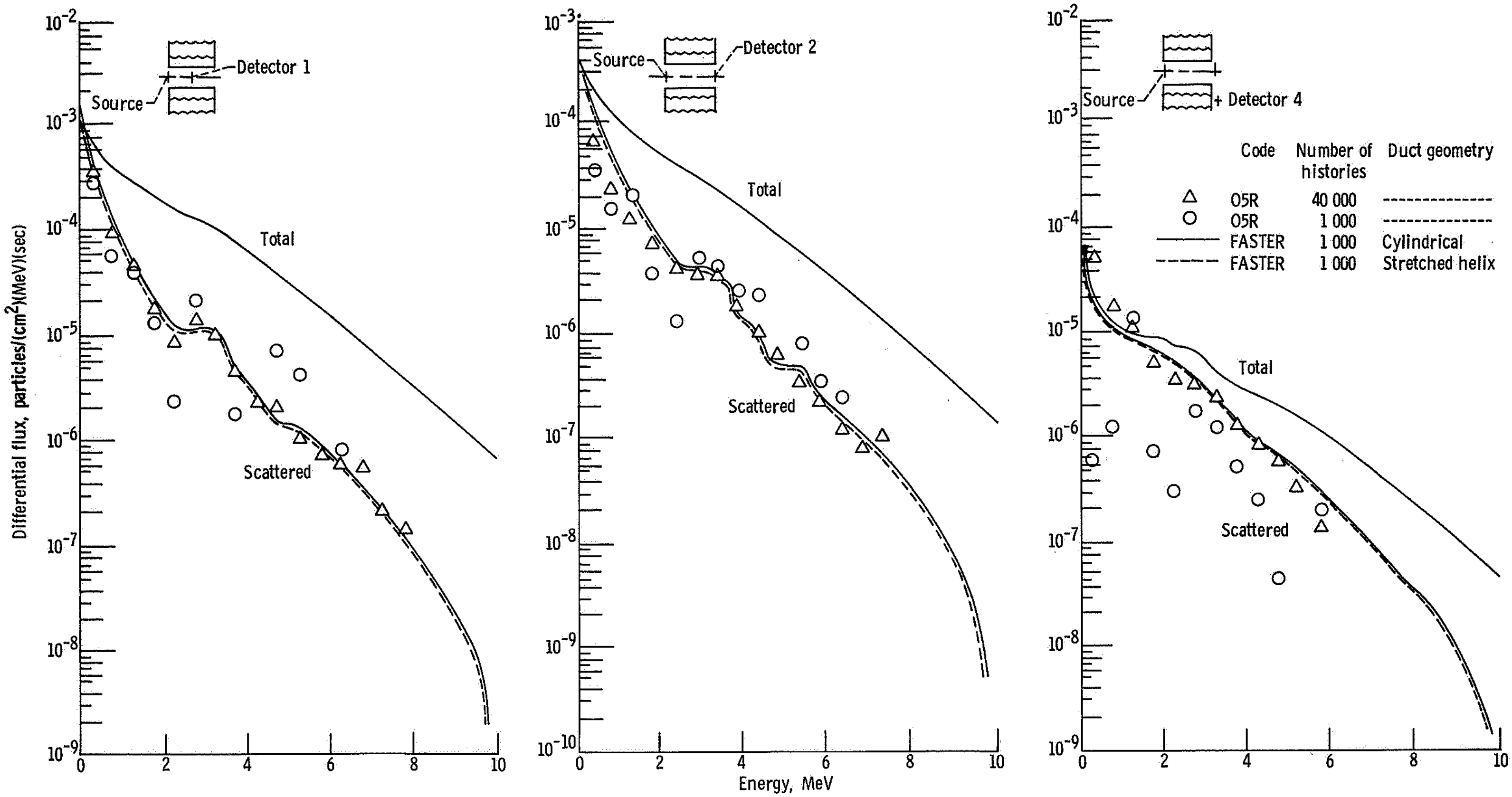


Figure 1. - Problem configuration.

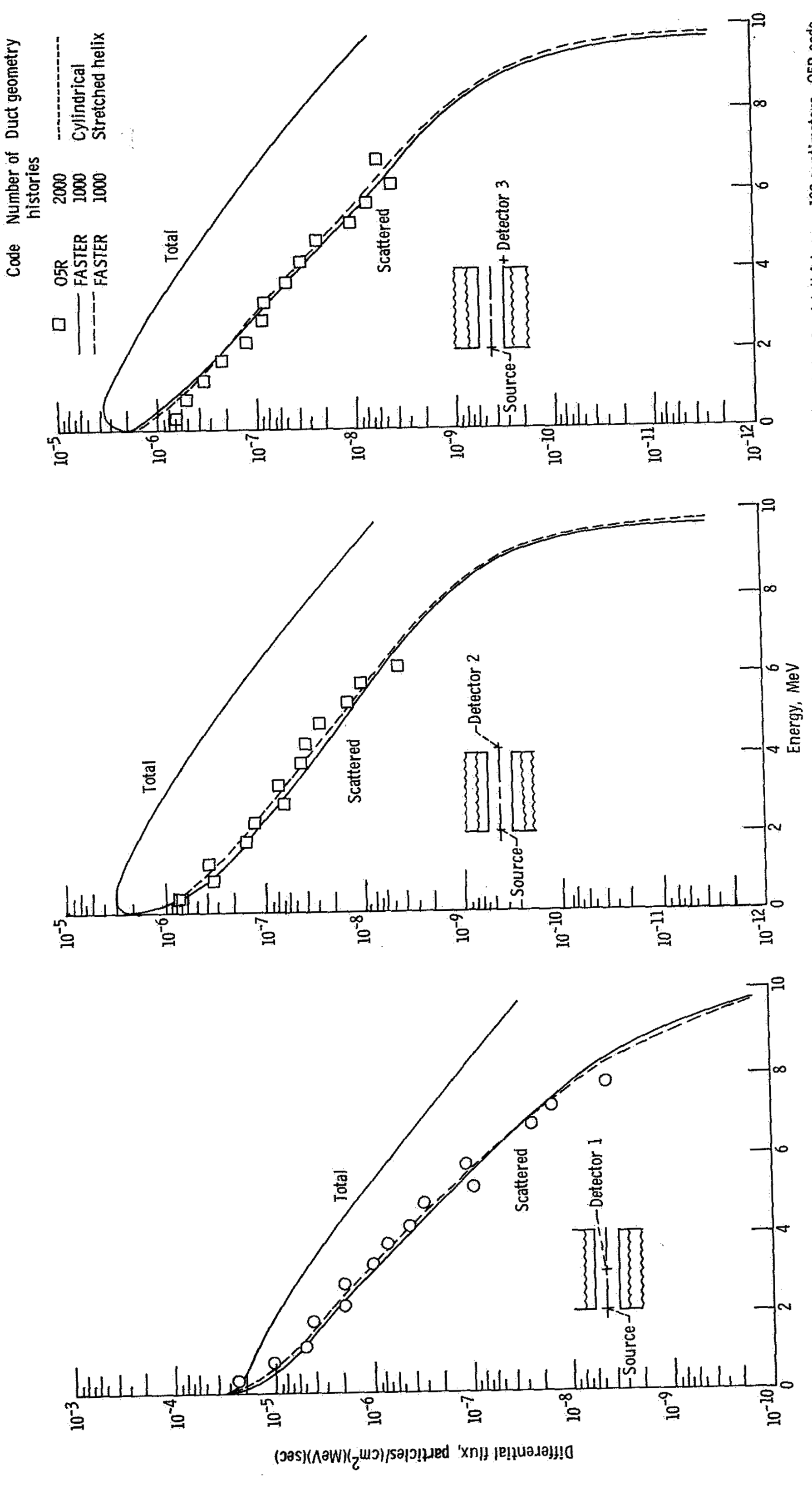


(a) Detector 1; slab thickness, 20 centimeters.

(b) Detector 2; slab thickness, 20 centimeters.

(c) Detector 4; slab thickness, 20 centimeters.

Figure 2. - Differential flux as function of energy.



(d) Detector 1; slab thickness, 100 centimeters; O5R code unbiased.

(e) Detector 2; slab thickness, 100 centimeters; O5R code biased.

(f) Detector 3; slab thickness, 100 centimeters; O5R code biased.

Figure 2. - Concluded.

NASA-Langley, 1969 —— 22 E-4976

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546