Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A05; Microfiche A01

ORNL/CSD/TM--185

DE83 000221

ORNL/CSD/TM-185 34A
Distribution Category UC ~~20c~~

MUXS: A CODE TO GENERATE MULTIGROUP CROSS SECTIONS FOR SPUTTERING CALCULATIONS

T. J. Hoffman, M. T. Robinson,* and H. L. Dodds, Jr.**

COMPUTER SCIENCES
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, TN 37830

Date Published - October 1982

*Solid State Division, Oak Ridge National Laboratory.

**Department of Nuclear Engineering. University of Tennessee, Knoxville, TN.

Union Carbide Corporation, Nuclear Division
operating the
Oak Ridge Gaseous Diffusion Plant
Oak Ridge Y-12 Plant
Paducah Gaseous Diffusion Plant
Oak Ridge National Laboratory
under Contract No. W-7405-eng-26
for the
DEPARTMENT OF ENERGY

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

The authors would like to thank D. K. Holmes for his help in implementing the cross section processing procedure from the MARLOWE code into MUXS. We would also like to acknowledge J. Bohdansky, Max-Planck-Institut fur Plasmaphysik, for valuable discussions regarding the experimental aspects of the sputtering and reflection data that was used to verify MUXS (see References 1-5). The efforts of Jeanette Hamby in preparing this manuscript are appreciated.

Research was sponsored by the Magnetic Fusion Energy Division, U. S. Department of Energy, under contract W-7405-eng-26 with the Union Carbide Corporation.

ABSTRACT

This report documents MUXS, a computer code to generate multigroup cross sections for charged particle transport problems. Cross sections generated by MUXS can be used in many multigroup transport codes, with minor modifications to these codes, to calculate sputtering yields, reflection coefficients, penetration distances, etc.

I. INTRODUCTION

Over the past few years we have presented several papers[1-5] on the application of neutral-particle, multigroup methods to the analysis of charged particle transport problems. The purpose of this report is to document the computer code that was developed during that period to generate multigroup cross sections. For more detail on the problems that were studied, the approach taken to these problems, and the results of our calculations, the reader is referred to the references.

Charged particle transport differs considerably from that for neutral particles. Unlike the nuclear forces that determine the directional changes of neutrons, the Coulomb forces that affect the paths of charged particles are extremely long range. This phenomenon leads to essentially continuous interactions between an ion and its environment of electrons and nuclei. Treatment of the many small deflections of the ion using a Legendre expansion of the differential scattering cross section, as is done for multigroup neutral particle transport, is simply not feasible; too many terms in the series would be required. The approach we have taken, and that used by others as well, is to account for the energy lost by the ion when deflected through small angles with a continuous slowing down term. This approach ignores the directional change due to these interactions. All electronic interactions with the ion are included in this term. Interactions with the nuclei that result in a scattering angle that is less than a prescribed angle θ_n are also treated as continuous interactions.

Scatterings through angles larger than θ_n are termed "discrete interactions" and are included in the evaluation of the multigroup total and scattering cross sections.

Continuous interactions lead to a term in the Boltzmann equation that is not present in the transport of neutral particles. Therefore, in addition to the use of MUXS, one must modify the transport code in order to perform charged particle calculations. For this reason, the modifications we made to ANISN[6] for the analysis of sputtering problems are described in Appendix B of this report.

The codes MUXS and ANISN can be obtained from:

Radiation Shielding Information Center
Oak Ridge National Laboratory
Building 6025
P. O. Box X
Oak Ridge, TN 37830.

II. MULTIGROUP PARAMETERS CALCULATED BY MUXS

As shown in Reference 1, the multigroup formulation of the sputtering problem can be written as follows:

$$\mu \frac{\partial \phi_G(x,\mu)}{\partial x} + \left(\Sigma_G^T + \Sigma_G^{CSD}\right) \phi_G(x,\mu)$$

$$= Q_G(x,\mu) + \Sigma_{G-1}^{CSD} \phi_{G-1}(x,\mu)$$

$$+ \sum_{G'=1}^{G} \int_{-1}^{1} \Sigma_{G'\to G}^{S}(\mu' \to \mu) \phi_{G'}(x,\mu') \, d\mu' \qquad (1)$$

for G = 1, 2, ...NTOT.

The first NI energy groups represent transport of the ions; the remaining NT=NTOT-NI are target atom energy groups.

The computer code MUXS generates the multigroup parameters Σ_G^T, Σ_G^{CSD}, and the Legendre coefficients of $\Sigma_{G'\to G}^S$. The actual solution to Eq. 1 is obtained with a modified version of ANISN that is discussed in Appendix B.

In the development of the multigroup equation, the flux is assumed separable within each energy group, i.e.,

$$\phi(x,E,\mu) = \phi_G(x,\mu)W_G(E) \text{ for } E\varepsilon(E_g, E_{g-1}) \tag{2}$$

With this assumption, the following definitions for the multigroup cross sections are obtained:

$$\Sigma_G^T = \frac{N \int_{E_g}^{E_{g-1}} \sigma_t(E)\ W_G(E)\ dE}{\int_{E_g}^{E_{g-1}} W_G(E)\ dE}, \tag{3}$$

$$\Sigma_G^{CSD} = \frac{S(E_g)\ W_G(E_g)}{\int_{E_g}^{E_{g-1}} W_G(E)\ dE}, \tag{4}$$

and

$$\Sigma^{S}_{G\to H}(\mu) = \frac{N \int_{E_g}^{E_{g-1}} W_G(E)\ \sigma[E\to(E_h,E_{h-1}),\mu]\ dE}{\int_{E_g}^{E_{g-1}} W_G(E)\ dE}, \qquad (5)$$

where

N = target atom number density.

The functions $\sigma_t(E)$, $S(E)$, $\sigma[E\to(E_h,E_{h-1},\mu]$, and $W_G(E)$ and their evaluation by MUXS, are described in the following sections.

a. The Point-Energy Total Cross Section

The point-energy total cross section, $\sigma_t(E)$ in Eq. 3, is obtained by integrating the differential cross section over all recoil particle energies that correspond to discrete interactions, i.e.,

$$\sigma_t(E) = \int_{T=\gamma_1 E}^{\gamma_2 E} d\sigma(E,T), \qquad (6)$$

where

$d\sigma(E,T)$ = differential cross section for the elastic scattering of an incident particle of energy E, producing a recoiling target atom with energy T,

$$\gamma_1 = \frac{\text{minimum recoil atom energy of a discrete interaction}}{\text{incident particle energy}},$$

$$= \frac{2m_1m_2}{(m_1+m_2)^2}\left\{1 + \frac{m_1}{m_2}\sin^2\theta_n - \cos\theta_n\left[1 - \left(\frac{m_1}{m_2}\right)^2 \sin^2\theta_n\right]^{1/2}\right\}, \qquad (7)$$

m_1 = mass of the incident particle,

m_2 = mass of the recoil atom,

θ_n = minimum scattering angle for a discrete interaction,

and

$$\gamma_2 = \frac{\text{maximum recoil atom energy}}{\text{incident particle energy}},$$

$$= \frac{4m_1m_2}{(m_1+m_2)^2}. \qquad (8)$$

Equations 7 and 8 are developed in Reference 1.

b. The Stopping Cross Section

The stopping cross section, S(E) in Eq. 4, is the sum of the stopping cross section for electronic interactions and that for elastic

interactions in which the incident particle scatters through an angle less than θ_n, i.e.,

$$S(E) = S_e(E) + N \int_{T=0}^{\gamma_1 E} T d\sigma(E,T). \qquad (9)$$

The stopping cross section for electronic interactions is calculated with the following formula from Lindhard and Scharff.[7]

$$S_e(E) = kN \left(\frac{Z_1 Z_2 e^2}{a_{12}}\right)^{1/2} (\pi a_{12}^2) \left(\frac{4m_1}{m_1+m_2}\right) \left(\frac{m_2}{m_1+m_2}\right)^{1/2} E^{1/2}, \qquad (10)$$

where

$$k = \frac{32}{3\pi} \left(\frac{m_e}{m_2}\right)^{1/2} \left(1 + \frac{m_2}{m_1}\right)^{3/2} \frac{Z_1^{2/3} Z_2^{1/2}}{(Z_1^{2/3}+Z_2^{2/3})^{3/4}}, \qquad (11)$$

$$a_{12} = \left(\frac{9\pi^2}{128}\right)^{1/3} a_o \left[Z_1^{2/3} + Z_2^{2/3}\right]^{-1/2} \qquad (12)$$

a_o = 0.529 Å, the Bohr radius,

m_e, e = mass in amu, charge of an electron,

m_1, Z_1 = mass in amu, atomic number of the incident particle,

m_2, Z_2 = mass in amu, atomic number of the recoil atom.

c. Calculation of the Legendre Coefficients

In Eq. 5, $\sigma[E \to E_h, E_{h-1}, \mu]$ represents the cross section for discrete interactions in which the incident particle of energy E scatters into

group H, i.e., (E_h, E_{h-1}). μ is the cosine of the scattering angle (laboratory). This cross section is expressed as a finite series of Legendre polynomials, $P_\ell(\mu)$, as follows:

$$\sigma[E \rightarrow (E_h, E_{h-1}), \mu] = \sum_{\ell=0}^{NCOEF-1} \sigma^{\ell}_{E\rightarrow IH} P_\ell(\mu), \tag{13}$$

where

$$\sigma^{\ell}_{E\rightarrow IH} = \frac{2\ell+1}{2} \int_{TMIN}^{TMAX} P_\ell(\mu)\, d\sigma(E,T) \tag{14}$$

For scatter from one ion group to another, this expression is

$$\sigma^{\ell}_{E\rightarrow IH} = \frac{2\ell+1}{2} \int_{Max(E-E_{h-1}, \gamma_1 E)}^{Min(E-E_h, \gamma_2 E)} P_\ell(\mu)\, d\sigma(E,T), \text{ for } IH \leq NI, \tag{15}$$

where

$$\mu = \left(1 - \frac{T}{E}\right)^{\frac{1}{2}} + \frac{1}{2}\left(1 - \frac{m_2}{m_1}\right)\left(\frac{T}{E}\right)\left(1 - \frac{T}{E}\right)^{-\frac{1}{2}} \tag{16}$$

The production of energetic target atoms as recoil atoms from ion scatterings is treated in a manner similar to the production of secondary gamma rays in neutron capture or inelastic scattering

reactions. However, unlike these neutron reactions, the recoil atom production is a highly anisotropic process. This anisotropy can be accounted for by treating these interactions in much the same manner one would treat the anisotropy of the emerging ion, i.e., with a Legendre expansion of the cosine of the angle between the incident particle direction and the emergent particle (recoil atom) direction. In the multigroup transport code, this process is treated as a "scatter" from an ion energy group into a target atom energy group. Therefore, transfer from an ion group into a target atom group is represented by the following Legendre coefficients:

$$\sigma^{\ell}_{E\to IH} = \frac{2\ell+1}{2} \int_{\mathrm{Max}(E_{h-1},\gamma_1 E)}^{\mathrm{Min}(E_h,\gamma_2 E)} P_\ell(\eta)\, d\sigma(E,T), \text{ for IH} > \text{NI}, \tag{17}$$

where

$$\eta = (T/\gamma_2 E)^{\frac{1}{2}}. \tag{18}$$

Equations 16 and 18 are obtained from the conservation of energy and of momentum.

For interactions between a target atom with energy E and a stationary target atom, either or both can emerge in energy group IH.

Therefore, the following expression is used to evaluate the Legendre coefficients for target atom interactions:

$$\sigma^{\ell}_{E\to IH} = \frac{2\ell+1}{2}\int_{\mathrm{Max}(E-E_{h-1},\gamma_1 E)}^{\mathrm{Min}(E-E_h,E)} P_\ell(\mu)\, d\sigma(E,T) + \frac{2\ell+1}{2}\int_{\mathrm{Max}(E_{h-1},\gamma_1 E)}^{\mathrm{Min}(E_h,E)} P_\ell(\eta)\, d\sigma(E,T), \qquad (19)$$

where

$$\mu = \left(1 - \frac{T}{E}\right)^{1/2}, \qquad (20)$$

and

$$\eta = \left(\frac{T}{E}\right)^{1/2}. \qquad (21)$$

d. Evaluation of the Weighting Function

Within each energy group, the energy-dependent particle flux is assumed to have the following functional form

$$\phi(E) = C_G E^{-XI_G} \text{ for } E_g < E < E_{g-1}. \qquad (22)$$

In MUXS, the group flux, ϕ_G, is assumed to be equal to $\phi(E)$ at the center of the energy interval, i.e.,

$$\phi\left(\frac{E_g+E_{g-1}}{2}\right) = \frac{\phi_G}{E_{g-1}-E_g}. \qquad (23)$$

It is also required that the energy dependent flux be continuous at the group boundary. This requirement is essential in order to conserve particles. Therefore, we require

$$C_G E_g^{-XI_G} = C_{G+1} E_g^{-XI_{G+1}}. \tag{24}$$

Equations 23 and 24 result in 2*NTOT-1 equations for 2*NTOT unknowns. The additional condition used is that $XI_1 = XI_2$. These requirements result in the following equations for the XI_G's:

$$XI_1 = \frac{\ell n \left[\frac{\phi_2}{\phi_1} \frac{(E_o - E_1)}{(E_1 - E_2)} \right]}{\ell n \left[\frac{E_o + E_1}{E_1 + E_2} \right]},$$

$$XI_G = \frac{XI_{G-1} \ell n \left(\frac{E_{g-2} + E_{g-1}}{2E_{g-1}} \right) - \ell n \left[\frac{\phi_{G+1}(E_{g-2} - E_{g-1})}{\phi_G (E_{g-1} - E_g)} \right]}{\ell n \left(\frac{E_{g-1} + E_g}{2E_{g-1}} \right)} \quad \text{for } G > 1. \tag{25}$$

Since the weighting function appears in both the numerator and denominator of the equations used by MUXS to evaluate the multigroup parameters, Eqs. 3-5, only XI_G is needed, i.e.,

$$W_G(E) \sim E^{-XI_G} \tag{26}$$

III. CALCULATIONAL ALGORITHM IN MUXS

The generation of the multigroup cross sections consists of the following steps:

1. The point-energy total cross section, $\sigma_t(E)$ of Eq. 3, and the P_o Legendre coefficient of the differential scattering cross section $\sigma[E \rightarrow E_h, E_{h-1}]$ of Eq. 5, are calculated at energies corresponding to the abscissa of a Gauss-Legendre quadrature set. The stopping cross section, S(E) of Eq. 4, is evaluated at the group boundaries. These cross sections are obtained at energy E by calling the following subroutines.

   a. Subroutine TSIGMA. This routine sets up a table of the differential scattering cross section, DSIG(E,T), as a function of T, the energy transferred to the recoiling target atom. The table is based on the approximation:

$$DSIG(T)/DSIG(TM) = (TM/T)^N,$$

   where TM is the maximum possible transferred energy. The tables contain values of $\ell n(TM/T)$ and of N. The values of DSIG are retrieved by the function DSIG(T).

b. Subroutine QUANC8. This routine integrates over T to determine the point cross sections using an adaptive quadrature. The adaptive quadrature ensures adequate accuracy in this precision-sensitive phase of the calculation.

2. The multigroup cross sections, SIG(G), are obtained from the point-energy cross sections, PSIG(E), using the following iterative procedure:

   a. An initial guess at the (infinite medium) group fluxes, PHIO(G), is made.

   b. Using PHIO(G), the energy-dependent flux, Eq. 22, is obtained

$$PHE(E) = C(G)*E**(-XI(G)),$$

   where C(G) and XI(G) are the group dependent parameters described in the previous section.

   c. The point cross sections are weighted by PHE(E) to obtain estimates of the multigroup cross sections, i.e.,

$$SIG(G) = \frac{\text{INTEGRAL OVER GROUP G OF PSIG(E)*PHE(E)}}{\text{INTEGRAL OVER GROUP G OF PHE(E)}}$$

   The integral in the numerator is obtained using a Gauss-Legendre quadrature; the PSIG(E)'s were evaluated at the abscissas for this reason. The denominator is calculated analytically.

d. With the estimated group cross sections, the infinite medium group fluxes, PHI(G), are calculated - see Reference 1.

e. The fractional change in the group flux, i.e., PHI(G) versus PHIO(G), is calculated. If this change is greater than 1% for any group, PHIO is set to PHI and MUXS returns to step b.

Note that this iterative procedure does not require reevaluation of the point-energy cross sections, a major time-consumer in MUXS.

3. The higher order Legendre coefficients for the multigroup scattering matrix, Eq. 5, are calculated.

4. An ANISN-formatted, binary cross section file for the multigroup cross section set is created. This file is suitable for input to many multigroup transport codes, e.g., ANISN,[6] DOT,[8] MORSE.[9] Or, alternatively, this file can be converted into an AMPX[10] working library by the computer code LAVA.[11]

IV. INPUT DESCRIPTION FOR MUXS

The input data requirements for MUXS are as follows:

1. Identification record: The first 48 characters of this record are printed on the output record to identify the calculations and are also passed to ANISN for identification of the cross section set.

2. The NAMELIST input record &DATA contains the remaining information required.

NI The number of ion groups.

NT The number of target atom groups. The total number of groups (NI + NT) must not exceed 50. NT can be set to zero for problems that do not require the transport of the target atoms, e.g., ion range calculations.

NCOEF The number of Legendre coefficients (must not exceed 20).

UO The minimum energy in eV that a target atom must have to be considered as sputtered. The cohesive energy is an appropriate choice.

DENSE The density of the target in atoms per cubic angstrom (or per barn-cm).

TH The minimum laboratory scattering angle for discrete interactions in degrees.

EI The initial kinetic energy of the incident ion in eV.

NEWS Controls the level of output from the program as follows:

NEWS=0 (The default value) causes printing of the input data, the converged fluxes, and the 1D multigroup cross sections.

NEWS=1 Causes the printing (in addition to the above) of the fluxes at each iteration.

NEWS=2 Causes the printing (in addition to both of the above) of the multigroup scattering matrix (Legendre coefficients).

NEWS=3 Causes the printing (in addition to all of the above) of all of the interpolation tables generated by Subroutine TSIGMA.

The following three variables are arrays of two quantities. The first element refers to the incident ion. The second to the target atoms:

AMU The mass of the particle in atomic mass units.

Z The atomic number of the particle.

BPAR The screening length for interactions of the particle with the target atoms in angstroms. If these parameters are omitted or entered as zero, the program will supply values using the Firsov formula.[12]

V. MUXS SAMPLE PROBLEM

As an example of a MUXS problem, the following data was input (unit 5) to obtain multigroup cross sections for an 8 kev proton incident on a nickel target:

```
SPUTTERING CROSS SECTIONS FOR H INCIDENT ON NI - 20GPS
 &DATA NI = 10, NT = 10, NCOEF = 20, UO = 4.458, TH = 2.5,
 DENSE = 0.0913, EI = 8000, Z = 1.0, 28.0, AMU = 1.0, 58.71,
 BPAR = 0.0, 0.075, NEWS = 0, &END
```

The printed output is shown in Fig. 1. The cross sections are stored by MUXS on unit 4. Calculation of sputtering yields with the cross sections produced in this MUXS sample problem is described in Appendix B.

SPUTTERING CROSS SECTIONS FOR N=>NI - 20GPS

10 ION GROUPS 10 TARGET ATOM GROUPS 20 LEGENDRE COEFFICIENTS

MINIMUM SCATTERING ANGLE (DEGREES) 2.5000000

TARGET BINDING ENERGY (E. V.) 4.4580000 TARGET DENSITY (/ANGSTROMS ** 3) .91300000D-01

INITIAL INCIDENT ION KINETIC ENERGY (E. V.) 8000.0000

INCIDENT ION ATOMIC NUMBER 1.0000000 ATOMIC MASS 1.0000000 SCREENING LENGTH (ANGSTROMS) .13746768

TARGET ATOM ATOMIC NUMBER 28.000000 ATOMIC MASS 58.710000 SCREENING LENGTH (ANGSTROMS) .75000000D-01

FLUXES AFTER ITERATION 9

IG	FLUX	IG	FLUX	IG	FLUX	IG	FLUX	IG	FLUX
1	166.46	2	165.65	3	164.66	4	163.37	5	161.71
6	159.68	7	156.79	8	152.69	9	147.35	10	138.79
11	1.7759	12	4.4386	13	7.1491	14	9.3275	15	11.271
16	12.854	17	14.235	18	15.416	19	16.459	20	17.377

GROUP 1	SIGT .44731371E-01	SIGSD .58527626E-02	S	84.864471
GROUP 2	SIGT .53468846E-01	SIGSD .58514439E-02	S	76.395035
GROUP 3	SIGT .65245509E-01	SIGSD .58472604E-02	S	67.921173
GROUP 4	SIGT .81591964E-01	SIGSD .58410652E-02	S	59.446594
GROUP 5	SIGT .10562923	SIGSD .58346502E-02	S	50.971115
GROUP 6	SIGT .13982838	SIGSD .58285855E-02	S	42.494751
GROUP 7	SIGT .19352[illegible]2	SIGSD .58034956E-02	S	34.020813
GROUP 8	SIGT .28095156	SIGSD .57893470E-02	S	25.540710

Figure 1. Printed Output from MUXS Sample Problem

GROUP 9	SIGT	.43571001	SIGSD	.57744682E-02	S 17.061432
GROUP 10	SIGT	.76317233	SIGSD	.[illegible]	S 8.5829124
GROUP 11	SIGT	.81756902	SIGSD	.30628532E-01	S 32.021469
GROUP 12	SIGT	1.0808506	SIGSD	.62245183E-01	S 21.323410
GROUP 13	SIGT	1.2865772	SIGSD	.98601043E-01	S 15.975158
GROUP 14	SIGT	1.4859104	SIGSD	.15019125	S 12.779972
GROUP 15	SIGT	1.6666307	SIGSD	.20868659	S 10.648841
GROUP 16	SIGT	1.8444185	SIGSD	.28026819	S 9.1262932
GROUP 17	SIGT	2.0193663	SIGSD	.36119848	S 7.9843016
GROUP 18	SIGT	2.1964455	SIGSD	.45370114	S 7.0960989
GROUP 19	SIGT	2.3763485	SIGSD	.55654293	S 6.3855524
GROUP 20	SIGT	2.5608454	SIGSD	.67018253	S 5.8042583

Figure 1 (contd.)

REFERENCES

1. T. J. Hoffman, H. L. Dodds, Jr., M. T. Robinson, and D. K. Holmes, Proceedings on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems, 2, 447 (1981).

2. T. J. Hoffman, M. T. Robinson, and H. L. Dodds, Jr., J. Nucl. Mater., 85, 597 (1979).

3. T. J. Hoffman, H. L. Dodds, Jr., M. T. Robinson, and D. K. Holmes, Nucl. Sci. Eng., 68, 204 (1978)

4. T. J. Hoffman, M. T. Robinson, H. L. Dodds, Jr., and D. K. Holmes, Trans. Am. Nucl. Soc., 34, 225 (1980).

5. T. J. Hoffman, H. L. Dodds, Jr., D. K. Holmes, and M. T. Robinson, Trans. Am. Nucl. Soc., 27, 266 (1977).

6. W. W. Engle, Jr., "A User's Manual for ANISN," K-1693, Oak Ridge Gaseous Diffusion Plant (1967).

7. J. Lindhard and M. Scharff, Phys. Rev., 124, 128 (1961).

8. W. A. Rhoades, et al., "The DOT-IV Two-Dimensional Discrete Ordinates Transport Code with Space-Dependent Mesh and Quadrature," ORNL/TM-6529, Oak Ridge National Laboratory (1979).

9. M. B. Emmett, "The MORSE Monte Carlo Radiation Transport Code System," ORNL-4972, Oak Ridge National Laboratory (1975).

10. N. M. Greene, J. L. Lucius, L. M. Petrie, W. E. Ford, III, J. E. White, and R. Q. Wright, "AMPX: A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Libraries from ENDF/B," ORNL/TM-3706, Oak Ridge National Laboratory (1976).

11. S. K. Fraley, "A User's Guide for LAVA," ORNL/CSD-11, Oak Ridge National Laboratory (1976).

12. O. B. Firsov, Zh. Eksp. Teor. Fiz. 33, 696 (1957). [Engl. transl.: Sov. Phys.-JETP, 6, 534 (1958)].

13. P. Sigmund, Phys. Rev., 184, 383 (1969).

14. J. Roth, J. Bohdansky, and W. Ottenberger, "Data on Low Energy Light Ion Sputtering," IPP9/26, Max-Planck-Institut fur Plasmaphysik (1979).

Appendix A

Listing of MUXS Program

```
C                                                                       MAIN0001
C                                                                       MAIN0002
C              'MUXS', A PROCEDURE FOR THE GENERATION OF                MAIN0003
C                                                                       MAIN0004
C        MULTIGROUP CROSS SECTIONS FOR SPUTTERING CALCULATIONS          MAIN0005
C                                                                       MAIN0006
C     USING THE MOLIERE APPROXIMATION TO THE THOMAS-FERMI POTENTIAL     MAIN0007
C                                                                       MAIN0008
C            TO DESCRIBE THE SCATTERING OF PROJECTILES AND              MAIN0009
C                                                                       MAIN0010
C              THE LSS ELECTRONIC STOPPING CROSS SECTION                MAIN0011
C                                                                       MAIN0012
C   REFERENCE:                                                          MAIN0013
C                                                                       MAIN0014
C       T. J. HOFFMAN, M. T. ROBINSON, AND H. L. DODDS, JR.,            MAIN0015
C       MUXS: A CODE TO GENERATE MULTIGROUP CROSS SECTIONS FOR          MAIN0016
C       SPUTTERING CALCULATIONS, ORNL/CSD/TM-185, OAK RIDGE             MAIN0017
C       NATIONAL LABORATORY (1982).                                     MAIN0018
C                                                                       MAIN0019
C                                                                       MAIN0020
C  INPUT DATA REQUIREMENTS (DATA SET REFERENCE NUMBER 5):               MAIN0021
C                                                                       MAIN0022
C     1. IDENTIFICATION RECORD:  THE FIRST 48 BYTES OF THIS RECORD ARE  MAIN0023
C        PRINTED ON THE OUTPUT RECORD TO IDENTIFY THE CALCULATIONS AND  MAIN0024
C        ARE ALSO PASSED TO 'ANISN' FOR THE SAME PURPOSE.               MAIN0025
C                                                                       MAIN0026
C     2. THE 'NAMELIST' INPUT RECORD '&DATA' CONTAINS THE REMAINING     MAIN0027
C        INFORMATION REQUIRED:                                          MAIN0028
C                                                                       MAIN0029
C        NI       THE NUMBER OF ION GROUPS                              MAIN0030
C                                                                       MAIN0031
C        NT       THE NUMBER OF TARGET ATOM GROUPS.  THE TOTAL NUMBER OF MAIN0032
C                 GROUPS (NI + NT) MUST NOT EXCEED 50.                  MAIN0033
C                                                                       MAIN0034
C        NCOEF    THE NUMBER OF LEGENDRE COEFFICIENTS.  MUST NOT EXCEED MAIN0035
C                 20.                                                   MAIN0036
C                                                                       MAIN0037
C        U0       THE MINIMUM ENERGY THAT A TARGET ATOM MUST HAVE TO BE MAIN0038
C                 CONSIDERED AS SPUTTERED.  THE COHESIVE ENERGY IS AN   MAIN0039
C                 APPROPRIATE CHOICE.  THE VALUE MUST BE IN E. V.       MAIN0040
C                                                                       MAIN0041
C        DENSE    THE DENSITY OF THE TARGET IN ATOMS PER CUBIC ANGSTROM MAIN0042
C                 (OR PER BARN-CM).                                     MAIN0043
C                                                                       MAIN0044
C        TH       THE MINIMUM LABORATORY SCATTERING ANGLE FOR DISCRETE  MAIN0045
C                 ATOMIC INTERACTIONS.                                  MAIN0046
C                                                                       MAIN0047
C        EI       THE INITIAL KINETIC ENERGY OF THE INCIDENT ION IN E. V. MAIN0048
C                                                                       MAIN0049
C        NEWS     CONTROLS THE LEVEL OF OUTPUT FROM THE PROGRAM, USING  MAIN0050
C                 THE CODE:                                             MAIN0051
C                                                                       MAIN0052
C                 NEWS = 0 (THE DEFAULT VALUE) CAUSES PRINTING OF THE   MAIN0053
C                          INPUT DATA, THE CONVERGED FLUXES, AND THE    MAIN0054
C                          1D MULTIGROUP CROSS SECTIONS ONLY.           MAIN0055
```

Appendix A (contd.)

```
C                                                                       MAIN0056
C              NEWS = 1 CAUSES THE PRINTING IN ADDITION OF THE FLUXES   MAIN0057
C                       AT EACH ITERATION.                              MAIN0058
C                                                                       MAIN0059
C              NEWS = 2 CAUSES THE PRINTING (IN ADDITION TO BOTH OF     MAIN0060
C                       THE ABOVE) OF THE MULTIGROUP TRANSFER MATRIX    MAIN0061
C                                                                       MAIN0062
C              NEWS = 3 CAUSES THE PRINTING (IN ADDITION TO ALL OF THE  MAIN0063
C                       ABOVE) OF ALL OF THE INTERPOLATION TABLES       MAIN0064
C                       GENERATED BY 'TSIG'.                            MAIN0065
C                                                                       MAIN0066
C  THE FOLLOWING THREE VARIABLES ARE ARRAYS OF TWO QUANTITIES.  THE     MAIN0067
C  FIRST ELEMENT REFERS TO THE INCIDENT ION, THE SECOND TO THE TARGET   MAIN0068
C  ATOMS:                                                               MAIN0069
C                                                                       MAIN0070
C       AMU     THE MASS OF THE PARTICLE IN ATOMIC MASS UNITS.          MAIN0071
C                                                                       MAIN0072
C       Z       THE ATOMIC NUMBER OF THE PARTICLE.                      MAIN0073
C                                                                       MAIN0074
C       BPAR    THE SCREENING LENGTH FOR INTERACTIONS OF THE PARTICLE   MAIN0075
C               WITH THE TARGET ATOMS, IN ANGSTROMS.  IF THESE PARAMET- MAIN0076
C               ERS ARE OMITTED OR ENTERED AS ZERO, THE PROGRAM WILL    MAIN0077
C               SUPPLY VALUES USING THE FIRSOV FORMULA.                 MAIN0078
C                                                                       MAIN0079
C                                                                       MAIN0080
C  OUTPUT DATA SETS:                                                    MAIN0081
C                                                                       MAIN0082
C       1. THE PRINTED OUTPUT IS DIRECTED TO THE STANDARD OUTPUT DATA   MAIN0083
C          SET ('FT06F001' IN IBM SYSTEMS).                             MAIN0084
C                                                                       MAIN0085
C       2. DATA SET REFERENCE NUMBER 4 IS USED TO STORE THE 'ANISN'-    MAIN0086
C          ORIENTED MULTIGROUP CROSS SECTIONS IN UNFORMATTED FORM.  IT  MAIN0087
C          IS ALSO USED FOR TEMPORARY STORAGE OF THE LEGENDRE COEFF-    MAIN0088
C          ICIENTS.                                                     MAIN0089
C                                                                       MAIN0090
C                                                                       MAIN0091
C  DESCRIPTION OF THE COMMON BLOCKS.  EXCEPT AS NOTED, ALL REAL         MAIN0092
C  VARIABLES ARE 'REAL*8'.                                              MAIN0093
C                                                                       MAIN0094
C     1. /PDAT/NI,NT,NCOEF,ION,DENSE,EI,UO,APQ(2),QGAM(2),RGAM(2),      MAIN0095
C              SGAM(2),GAMMA(2),TLOW(2),TLOG(2),APAR(2),BPAR(2),        MAIN0096
C              CPAR(2),ESTOP(2),NEWS,NTOT                               MAIN0097
C                                                                       MAIN0098
C       ION     USED TO INDICATE THE CURRENT PROJECTILE ( = 1 FOR THE   MAIN0099
C               INCIDENT ION, = 2 FOR THE TARGET ATOM).                 MAIN0100
C                                                                       MAIN0101
C       APQ     THESE FIVE ARRAYS CONTAIN VARIOUS FUNCTIONS OF THE      MAIN0102
C       QGAM    MASSES OF THE TWO PARTICLES.  EACH ARRAY CONTAINS TWO   MAIN0103
C       RGAM    VALUES, ONE FOR EACH PARTICLE AS PROJECTILE.            MAIN0104
C       SGAM                                                            MAIN0105
C       GAMMA                                                           MAIN0106
```

Appendix A (contd.)

```
C                                                                       MAIN0107
C       TLOW    THESE TWO ARRAYS DESCRIBE THE MINIMUM VALUES OF THE     MAIN0108
C       TLOG    PROJECTILE KINETIC ENERGIES FOR WHICH DISCRETE INTER-   MAIN0109
C               ACTIONS ARE USED, ONE VALUE FOR EACH PROJECTILE.        MAIN0110
C                                                                       MAIN0111
C       APAR    MOLIERE POTENTIAL PARAMETERS, ONE VALUE FOR EACH        MAIN0112
C       CPAR    PROJECTILE.                                             MAIN0113
C                                                                       MAIN0114
C       ESTOP   THE PARAMETERS OF THE LSS INELASTIC ENERGY LOSS MODEL.  MAIN0115
C                                                                       MAIN0116
C       NTOT    NI + NT, THE TOTAL NUMBER OF ENERGY GROUPS.             MAIN0117
C                                                                       MAIN0118
C                                                                       MAIN0119
C     2. /WATE/U(4),W(4),AB(10),WT1(10),INP,MEHL                        MAIN0120
C                                                                       MAIN0121
C       U, W    THE ABSCISSAS AND WEIGHTS FOR THE GAUSS-MEHLER QUADRAT- MAIN0122
C               URE USED TO EVALUATE THE CLASSICAL SCATTERING INTEGRALS MAIN0123
C               IN THE SUBROUTINE 'TSIGMA'.                             MAIN0124
C                                                                       MAIN0125
C       AB, WT1 THE ABSCISSAS AND WEIGHTS FOR THE GAUSS-LEGENDRE QUAD-  MAIN0126
C               RATURE USED TO EVALUATE THE POINT-ENERGY LEGENDRE       MAIN0127
C               COEFFICIENTS IN THE SUBROUTINE 'LEGEND' AND FOR         MAIN0128
C               INTEGRATIONS OVER ENERGY IN THE SUBROUTINE 'CROSSX'.    MAIN0129
C                                                                       MAIN0130
C       INP     THE NUMBER OF POINTS IN THE GAUSS-LEGENDRE SET.         MAIN0131
C                                                                       MAIN0132
C       MEHL    THE NUMBER OF POINTS IN THE GAUSS-MEHLER SET.           MAIN0133
C                                                                       MAIN0134
C                                                                       MAIN0135
C     3. /SIGX/EBI(51),EBT(51),S(50),SIGT(50),SIGSD(50),SIGS(50,50,20)  MAIN0136
C                                                                       MAIN0137
C       EBI     ENERGY GROUP BOUNDARIES FOR THE ION GROUPS.             MAIN0138
C                                                                       MAIN0139
C       EBT     ENERGY GROUP BOUNDARIES FOR THE TARGET ATOM GROUPS.     MAIN0140
C                                                                       MAIN0141
C       S       STOPPING CROSS SECTION AT THE GROUP BOUNDARIES FOR THE  MAIN0142
C               CONTINUOUS INTERACTIONS (REAL*4).                       MAIN0143
C                                                                       MAIN0144
C       SIGT    TOTAL GROUP CROSS SECTION FOR DISCRETE INTERACTIONS     MAIN0145
C               (REAL*4).                                               MAIN0146
C                                                                       MAIN0147
C       SIGSD   CONTINUOUS SLOWING DOWN GROUP CROSS SECTION (REAL*4).   MAIN0148
C                                                                       MAIN0149
C       SIGS    GROUP SCATTERING CROSS SECTION MATRIX FOR DISCRETE      MAIN0150
C               INTERACTIONS (REAL*4).                                  MAIN0151
C                                                                       MAIN0152
C                                                                       MAIN0153
C     4. /TABL/TMX,SIGO,KM,KQ,TE(200),SIG(200)                          MAIN0154
C                                                                       MAIN0155
C       TMX     THE RECIPROCAL OF THE MAXIMUM POSSIBLE TRANSFERRED      MAIN0156
C               ENERGY, TM.                                             MAIN0157
C                                                                       MAIN0158
```

Appendix A (contd.)

```
C       SIGO    THE DIFFERENTIAL CROSS SECTION FOR TRANSFERRING 'TM'.   MAIN0159
C                                                                       MAIN0160
C       KM      THE NUMBER OF ENTRIES IN THE INTERPOLATION TABLES.      MAIN0161
C                                                                       MAIN0162
C       KQ      KM - 1.                                                 MAIN0163
C                                                                       MAIN0164
C       TE      LN( TM / T ) AT THE TABULAR POINTS, WHERE T IS THE      MAIN0165
C               TRANSFERRED ENERGY.                                     MAIN0166
C                                                                       MAIN0167
C       SIG     LN( SIGMA / SIGO ) / LN( TM / T ), AT THE TABULAR       MAIN0168
C               POINTS, WHERE SIGMA IS THE DIFFERENTIAL CROSS SECTION   MAIN0169
C               FOR TRANSFERRING T.                                     MAIN0170
C                                                                       MAIN0171
C                                                                       MAIN0172
C  ERROR TERMINATIONS:                                                  MAIN0173
C                                                                       MAIN0174
C  THE PROGRAM ALWAYS TERMINATES WITH 'STOP N', WHERE THE VALUE OF      MAIN0175
C  'N' IDENTIFIES THE TERMINAL CONDITION AS FOLLOWS:                    MAIN0176
C                                                                       MAIN0177
C       N =  0  NORMAL TERMINATION AT THE END OF THE PROGRAM.           MAIN0178
C                                                                       MAIN0179
C       N =  8  STORAGE LIMITS FOR THE INTERPOLATION TABLES HAVE BEEN   MAIN0180
C               EXCEEDED IN 'TSIGMA'.  INCREASE THE DIMENSIONS IN THE   MAIN0181
C               COMMON BLOCK 'TABL'.                                    MAIN0182
C                                                                       MAIN0183
C       N =  9  AN OVERFLOW HAS BEEN ENCOUNTERED IN 'DSIG/TDSIG'.       MAIN0184
C                                                                       MAIN0185
C       N = 10, 11, ..., 17  FAILURE OF CONVERGENCE OF THE INTEGRATION  MAIN0186
C               PROCEDURE 'QUANC8', CALLED AT SEVERAL POINTS IN         MAIN0187
C               'CROSSX'.                                               MAIN0188
C                                                                       MAIN0189
C       N = 18  FAILURE OF CONVERGENCE IN DETERMINING THE GAUSS -       MAIN0190
C               LEGENDRE PARAMETERS IN 'IMTQL2', CALLED FROM 'GAUSS',   MAIN0191
C               CALLED IN TURN FROM 'MAIN'.                             MAIN0192
C                                                                       MAIN0193
C  VARIABLE SPECIFICATIONS:                                             MAIN0194
      IMPLICIT REAL*8 (A - H, O - Z)                                    MAIN0195
      INTEGER*4 NAME(12)                                                MAIN0196
      REAL*8 AMU(2),Z(2),GWS(10)                                        MAIN0197
      REAL*4 SIGT,SIGS,SIGSD,S,SPUN(2650)                               MAIN0198
      COMMON/PDAT/NI,NT,NCOEF,ION,DENSE,EI,UO,APQ(2),QGAM(2),RGAM(2),   MAIN0199
     1            SGAM(2),GAMMA(2),TLOW(2),TLOG(2),APAR(2),BPAR(2),   MAIN0200
     2            CPAR(2),ESTOP(2),NEWS,NTOT                          MAIN0201
      COMMON/SIGX/EBI(51),EBT(51),S(50),SIGT(50),SIGSD(50),             MAIN0202
     1            SIGS(50,50,20)                                      MAIN0203
      COMMON/WATE/U(4),W(4),AB(10),WT1(10),INP,MEHL                     MAIN0204
C                                                                       MAIN0205
C  DEFINITION OF NAMELIST INPUT RECORD:                                 MAIN0206
      NAMELIST/DATA/NEWS,NI,NT,NCOEF,UO,DENSE,TH,EI,AMU,Z,BPAR          MAIN0207
C                                                                       MAIN0208
C  FORMAT FOR IDENTIFICATION RECORD:                                    MAIN0209
   10 FORMAT(12A4)                                                      MAIN0210
```

Appendix A (contd.)

```
C                                                                       MAIN0211
C  DEFAULT VALUES OF THE INPUT DATA:                                    MAIN0212
      NEWS = 0                                                          MAIN0213
      BPAR(1) = 0D0                                                     MAIN0214
      BPAR(2) = 0D0                                                     MAIN0215
C                                                                       MAIN0216
C  READ THE CROSS SECTION TITLE:                                        MAIN0217
      READ(5, 10) NAME                                                  MAIN0218
C                                                                       MAIN0219
C  READ 'NAMELIST' INPUT:                                               MAIN0220
      READ( 5, DATA)                                                    MAIN0221
C                                                                       MAIN0222
C   GENERATE QUADRATURE SETS                                            MAIN0223
      INP = 10                                                          MAIN0224
      MEHL = 4                                                          MAIN0225
      CALL MEHLER( MEHL, QA, U, W)                                      MAIN0226
      CALL GAUSS( INP, AB, WT1, GWS)                                    MAIN0227
      NTOT = NI + NT                                                    MAIN0228
C                                                                       MAIN0229
C  CONSTRUCT MASS FACTORS:                                              MAIN0230
      QA = AMU(2) / AMU(1)                                              MAIN0231
      QB = 1D0 + QA                                                     MAIN0232
      APQ(1) = QA / QB                                                  MAIN0233
      QGAM(1) = 5D-1 * QB                                               MAIN0234
      RGAM(1) = 5D-1 * QGAM(1)                                          MAIN0235
      GAMMA(1) = APQ(1) / RGAM(1)                                       MAIN0236
      SGAM(1) = 1D0 / DSQRT( QA )                                       MAIN0237
      APQ(2) = 5D-1                                                     MAIN0238
      RGAM(2) = 5D-1                                                    MAIN0239
      GAMMA(2) = 1D0                                                    MAIN0240
      QGAM(2) = 1D0                                                     MAIN0241
      SGAM(2) = 1D0                                                     MAIN0242
      TLOW(2) = DSIN( 1.7453292519943296D-2 * TH ) ** 2                 MAIN0243
      QB = QA * QA - TLOW(2)                                            MAIN0244
      IF( QB .GT. 0D0 ) GO TO 20                                        MAIN0245
      QB = 0D0                                                          MAIN0246
      GO TO 30                                                          MAIN0247
   20 QB = DSQRT( QB * ( 1D0 - TLOW(2) ) )                              MAIN0248
   30 TLOW(1) = ( QA + TLOW(2) - QB ) / ( QA + QA )                     MAIN0249
      TLOG(1) = - DLOG( TLOW(1) )                                       MAIN0250
      TLOG(2) = - DLOG( TLOW(2) )                                       MAIN0251
      TLOW(1) = GAMMA(1) * TLOW(1)                                      MAIN0252
C                                                                       MAIN0253
C  EVALUATE THE MOLIERE POTENTIAL PARAMETERS:                           MAIN0254
      IF( BPAR(1) .LE. 0D0 ) BPAR(1) = 0.468493440443D0 *               MAIN0255
     1          DEXP( - DLOG( DSQRT( Z(1) ) + DSQRT( Z(2) ) ) / 1.5D0 ) MAIN0256
      IF( BPAR(2) .LE. 0D0 ) BPAR(2) = 0.468493440443D0 *               MAIN0257
     1          DEXP( - ( 1.3862943611198906D0 + DLOG( Z(2) ) ) / 3D0 ) MAIN0258
      APAR(1) = 5.039645408D0 * Z(1) * Z(2)                             MAIN0259
      APAR(2) = 5.039645408D0 * Z(2) * Z(2)                             MAIN0260
      CPAR(1) = 0.35D0 * BPAR(1) / APAR(1)                              MAIN0261
      CPAR(2) = 0.35D0 * BPAR(2) / APAR(2)                              MAIN0262
```

Appendix A (contd.)

```
C                                                                       MAIN0263
C  EVALUATE LSS INELASTIC STOPPING PARAMETERS:                          MAIN0264
      QA = DEXP( DLOG( Z(1) ) / 3D0 )                                   MAIN0265
      QB = DEXP( DLOG( Z(2) ) / 3D0 )                                   MAIN0266
      QC = DEXP( 1.5D0 * DLOG( QA ** 2 + QB ** 2 ) )                    MAIN0267
      ESTOP(1) = 1.2166547D0 * Z(1) * Z(2) * DSQRT( QA / AMU(1) ) / QC  MAIN0268
      ESTOP(2) = 0.4301524&D0 * Z(2) * DSQRT( QB / AMU(2) )             MAIN0269
C                                                                       MAIN0270
C  REPORT THE INPUT PARAMETERS:                                         MAIN0271
      PRINT 100, NAME, NI, NT, NCOEF, TH, U0, DENSE, EI,                MAIN0272
     1           (Z(IH), AMU(IH), BPAR(IH), IH = 1, 2)                  MAIN0273
C                                                                       MAIN0274
C  EVALUATE THE REQUIRED CROSS SECTIONS:                                MAIN0275
      CALL CROSSX                                                       MAIN0276
      DO 50 L = 1, NCOEF                                                MAIN0277
      LM1 = L - 1                                                       MAIN0278
      IF( L .LE. 1 )                                                    MAIN0279
     1       PRINT 110, (IG, SIGT(IG), SIGSD(IG), S(IG), IG = 1, NTOT)  MAIN0280
      IF( NEWS .LE. 2 ) GO TO 60                                        MAIN0281
C                                                                       MAIN0282
C  IF REQUESTED, PRINT MULTIGROUP TRANSFER MATRIX                       MAIN0283
      DO 40 IG = 1, NTOT                                                MAIN0284
      IF( IG .EQ. 1 ) PRINT 120, LM1                                    MAIN0285
   40 PRINT 130, IG, NTOT, (SIGS(IH,IG,L), IH = 1, NTOT)                MAIN0286
   50 CONTINUE                                                          MAIN0287
C                                                                       MAIN0288
C   CROSS SECTIONS ARE WRITTEN ON UNIT 4 (ANISN-FORMAT)                 MAIN0289
   60 REWIND 4                                                          MAIN0290
      IHM = NTOT + 3                                                    MAIN0291
      NPUN = IHM * NTOT                                                 MAIN0292
      DO 90 L = 1, NCOEF                                                MAIN0293
      DO 80 IG = 1, NTOT                                                MAIN0294
      IND = IHM * (IG-1) + 1                                            MAIN0295
      SPUN(IND) = 0.0                                                   MAIN0296
      IND = IND + 1                                                     MAIN0297
      SPUN(IND) = 0.0                                                   MAIN0298
      IND = IND + 1                                                     MAIN0299
      SPUN(IND) = 0.0                                                   MAIN0300
      IF( L .EQ. 1 ) SPUN(IND) = SIGT(IG)                               MAIN0301
      DO 70 IH = 1, NTOT                                                MAIN0302
      IND = IND + 1                                                     MAIN0303
      SPUN(IND) = 0.0                                                   MAIN0304
      IGR = IG - IH + 1                                                 MAIN0305
      IF( IGR .GT. 0 ) SPUN(IND) = SIGS(IGR,IG,L)                       MAIN0306
   70 CONTINUE                                                          MAIN0307
   80 CONTINUE                                                          MAIN0308
      IDENT = 0                                                         MAIN0309
      WRITE(4) NTOT, IHM, IDENT, L, NAME                                MAIN0310
   90 WRITE(4) (SPUN(IH), IH = 1, NPUN)                                 MAIN0311
C                                                                       MAIN0312
C   OUTPUT INDICATOR (TO ANISN) THAT LAST MATERIAL HAS BEEN READ        MAIN0313
      IDENT = 7                                                         MAIN0314
      WRITE(4) NTOT, IHM, IDENT, L, NAME                                MAIN0315
```

Appendix A (contd.)

```
C                                                                       MAIN0316
C   ADD ON SOME USEFUL INFORMATION FOR SPUTTERING CALCULATION           MAIN0317
      WRITE(4) (SIGSD(IH), IH = 1, NTOT)                                MAIN0318
      NTP1 = NT + 1                                                     MAIN0319
      WRITE(4) (EBT(IH), IH = 1, NTP1)                                  MAIN0320
C                                                                       MAIN0321
C  NORMAL PROGRAM TERMINATION:                                          MAIN0322
      STOP 0                                                            MAIN0323
C                                                                       MAIN0324
C  OUTPUT FORMATS:                                                      MAIN0325
  100 FORMAT('1',42X,12A4//25X,I3,' ION GROUPS',10X,I3,' TARGET ATOM GROMAIN0326
     1UPS',10X,I3,' LEGENDRE COEFFICIENTS'//41X,'MINIMUM SCATTERING ANGLMAIN0327
     2E (DEGREES)',G15.8//15X,'TARGET BINDING ENERGY (E. V.)',G15.8,10X,MAIN0328
     3'TARGET DENSITY (/ANGSTROMS ** 3)',G15.8//37X,'INITIAL INCIDENT IOMAIN0329
     4N KINETIC ENERGY (E. V.)',G15.8//' INCIDENT ION ATOMIC NUMBER',G15MAIN0330
     5.8,10X,'ATOMIC MASS',G15.8,10X,'SCREENING LENGTH (ANGSTROMS)',G15.MAIN0331
     68//' TARGET ATOM  ATOMIC NUMBER',G15.8,10X,'ATOMIC MASS',G15.8,10XMAIN0332
     7,'SCREENING LENGTH (ANGSTROMS)',G15.8)                            MAIN0333
  110 FORMAT(//19X,'GROUP',I3,10X,'SIGT',G15.8,10X,'SIGSD',G15.8,10X,'S'MAIN0334
     1,G15.8)                                                           MAIN0335
  120 FORMAT(///' P',I2,'  COEFFICIENT')                                MAIN0336
  130 FORMAT(/5X,'SIGS(IH ->',I3,'), IH = 1,',I3,')',/(10X,8G14.6))     MAIN0337
      END                                                               MAIN0338
C                                                                       CRSX0001
      SUBROUTINE CROSSX                                                 CRSX0002
C  EVALUATES THE MULTIGROUP CROSS SECTIONS FROM WHICH 'ANISN' CAN       CRSX0003
C  PERFORM SPUTTERING AND OTHER ATOMIC TRANSPORT CALCULATIONS.          CRSX0004
C                                                                       CRSX0005
C  SOME VARIABLE DEFINITIONS:                                           CRSX0006
C                                                                       CRSX0007
C       PSIGT   POINT ENERGY TOTAL CROSS SECTION FOR DISCRETE INTER-    CRSX0008
C               ACTIONS.                                                CRSX0009
C                                                                       CRSX0010
C       PSIGS   POINT ENERGY SCATTERING CROSS SECTION FOR DISCRETE      CRSX0011
C               INTERACTIONS.                                           CRSX0012
C                                                                       CRSX0013
C       COEF    POINT ENERGY LEGENDRE COEFFICIENTS. THESE COEFFICIENTS  CRSX0014
C               EVALUATED IN SUBROUTINE 'LEGEND' AND STORED ON DATA SET CRSX0015
C               REFERENCE NUMBER 4.  THEY ARE RECOVERED AND USED TO     CRSX0016
C               CALCULATE THE HIGHER ORDER MULTIGROUP LEGENDRE COEFF-   CRSX0017
C               CIENTS AFTER CONVERGENCE OF THE INFINITE MEDIUM FLUXES. CRSX0018
C                                                                       CRSX0019
C       PNORM   USED TO RENORMALIZE THE HIGHER ORDER COEFFICIENTS TO    CRSX0020
C               THE P0 COEFFICIENT.                                     CRSX0021
C                                                                       CRSX0022
C                                                                       CRSX0023
C  TO OBTAIN THE MULTIGROUP CROSS SECTIONS, THE POINT CROSS SECTIONS    CRSX0024
C  ARE WEIGHTED BY:                                                     CRSX0025
C                                                                       CRSX0026
C                PHE(E) = C(G) * E ** ( - XI(G) ),                      CRSX0027
C                                                                       CRSX0028
```

Appendix A (contd.)

```
C  WHERE C(G) AND XI(G) ARE GROUP DEPENDENT.  XI IS OBTAINED FROM THE   CRSX0029
C  INFINITE MEDIUM GROUP FLUX, PHI(G).  DEN(G) IS THE INTEGRAL OF       CRSX0030
C  PHE(E) OVER THE ENERGY GROUP.  THE GROUP FLUX IS OBTAINED            CRSX0031
C  ITERATIVELY.  PHIO(G) IS THE GROUP FLUX FROM THE PREVIOUS ITERATION. CRSX0032
C  IT IS USED TO DETERMINE CONVERGENCE.                                 CRSX0033
C                                                                       CRSX0034
C  VARIABLE SPECIFICATIONS:                                             CRSX0035
      IMPLICIT REAL*8 (A - H, O - Z)                                    CRSX0036
      EXTERNAL TDSIG,DSIG                                               CRSX0037
      REAL*8 COEF(20),XI(50),PHI(50),DEN(50),PHIO(50),E(50,10),         CRSX0038
     1       WTS(50,10),PSIGT(50,10),PSIGS(50,50,10),PNORM(50,50)       CRSX0039
      REAL*4 SIGT,SIGS,SIGSD,S                                          CRSX0040
      COMMON/PDAT/NI,NT,NCOEF,ION,DENSE,EI,U0,APQ(2),QGAM(2),RGAM(2),   CRSX0041
     1            SGAM(2),GAMMA(2),TLOW(2),TLOG(2),APAR(2),BPAR(2),     CRSX0042
     2            CPAR(2),ESTOP(2),NEWS,NTOT                            CRSX0043
      COMMON/SIGX/EBI(51),EBT(51),S(50),SIGT(50),SIGSD(50),             CRSX0044
     1            SIGS(50,50,20)                                        CRSX0045
      COMMON/WATE/U(4),W(4),AB(10),WT1(10),INP,MEHL                     CRSX0046
C                                                                       CRSX0047
C  GENERATE GROUP BOUNDARIES AND MAKE INITIAL FLUX GUESSES:             CRSX0048
      NIP1 = NI + 1                                                     CRSX0049
      NIP2 = NIP1 + 1                                                   CRSX0050
      NTP1 = NT + 1                                                     CRSX0051
      L = 1                                                             CRSX0052
      IF( NT .EQ. 0 ) L = 2                                             CRSX0053
      EBI(1) = EI                                                       CRSX0054
      EBI(NIP1) = U0 / GAMMA(L)                                         CRSX0055
      A = ( DSQRT( EBI(1) ) - DSQRT( EBI(NIP1) ) ) / NI                 CRSX0056
      PHIO(1) = A                                                       CRSX0057
      DO 10 IG = 2, NI                                                  CRSX0058
      PHIO(IG) = A                                                      CRSX0059
   10 EBI(IG) = ( DSQRT( EBI(IG-1) ) - A ) ** 2                         CRSX0060
      IF( NT .EQ. 0 ) GO TO 30                                          CRSX0061
      EBT(1) = EI * GAMMA(1)                                            CRSX0062
      EBT(NTP1) = U0                                                    CRSX0063
      QQ = ( 1D0 / DSQRT( EBT(NTP1) ) - 1D0 / DSQRT( EBT(1) ) ) / NT    CRSX0064
      PHIO(NIP1) = QQ                                                   CRSX0065
      DO 20 IG = 2, NT                                                  CRSX0066
      PHIO(NI+IG) = QQ                                                  CRSX0067
   20 EBT(IG) = ( 1D0 / ( QQ + 1D0 / DSQRT( EBT(IG-1) ) ) ** 2 )        CRSX0068
C                                                                       CRSX0069
C   DETERMINE POINT ENERGIES:                                           CRSX0070
   30 DO 50 IG = 1, NI                                                  CRSX0071
      A = 0.5D0 * ( EBI(IG) - EBI(IG+1) )                               CRSX0072
      QQ = 0.5D0 * ( EBI(IG) + EBI(IG+1) )                              CRSX0073
      DO 40 NN = 1, INP                                                 CRSX0074
   40 E(IG,NN) = A * AB(NN) + QQ                                        CRSX0075
   50 CONTINUE                                                          CRSX0076
      DO 70 IG = 1, NT                                                  CRSX0077
      A = 0.5D0 * ( EBT(IG) - EBT(IG+1) )                               CRSX0078
      QQ = 0.5D0 * ( EBT(IG) + EBT(IG+1) )                              CRSX0079
      DO 60 NN = 1, INP                                                 CRSX0080
   60 E(NI+IG,NN) = A * AB(NN) + QQ                                     CRSX0081
   70 CONTINUE                                                          CRSX0082
```

Appendix A (contd.)

```
C                                                                       CRSX0083
C  END OF INITIALIZATION.                                              CRSX0084
C                                                                       CRSX0085
C************BEGIN ION POINT-ENERGY CROSS SECTION EVALUATION**********CRSX0086
C                                                                       CRSX0087
C  SET ION FLAG AND START LOOP OVER ALL ION GROUPS:                     CRSX0088
      ION = 1                                                           CRSX0089
      DO 110 IG = 1, NI                                                 CRSX0090
C                                                                       CRSX0091
C  CALCULATE STOPPING CROSS SECTION AT LOWER ENERGY GROUP BOUNDARIES:   CRSX0092
      CALL TSIGMA( EBI(IG+1) )                                          CRSX0093
      TMIN = 0D0                                                        CRSX0094
      TMAX = TLOW(1) * EBI(IG+1)                                        CRSX0095
      CALL QUANC8( TDSIG, TMIN, TMAX, 0D0, 1D-4, ELAS, ERREST,          CRSX0096
     1                                                  NOFUN, FLAG)    CRSX0097
      IF( FLAG .NE. 0D0 ) STOP 10                                       CRSX0098
      S(IG) = ELAS + ESTOP(1) * DSQRT( EBI(IG+1) )                      CRSX0099
C                                                                       CRSX0100
C  EVALUATE TOTAL POINT ENERGY CROSS SECTION FOR DISCRETE INTERACTIONS: CRSX0101
      DO 100 NN = 1, INP                                                CRSX0102
      CALL TSIGMA( E(IG,NN) )                                           CRSX0103
      TMIN = TLOW(1) * E(IG,NN)                                         CRSX0104
      TMAX = GAMMA(1) * E(IG,NN)                                        CRSX0105
      CALL QUANC8( DSIG, TMIN, TMAX, 0D0, 1D-5, TOT, ERREST,            CRSX0106
     1                                                  NOFUN, FLAG)    CRSX0107
      IF( FLAG .NE. 0D0 ) STOP 11                                       CRSX0108
      PSIGT(IG,NN) = TOT                                                CRSX0109
C                                                                       CRSX0110
C  CALCULATE POINT-ENERGY SCATTERING CROSS SECTION FOR INTERACTIONS IN  CRSX0111
C  WHICH THE INCIDENT ION EMERGES IN GROUP IH:                          CRSX0112
      DO 80 IH = IG, NI                                                 CRSX0113
      TMIN = DMAX1( E(IG,NN) - EBI(IH), TLOW(1) * E(IG,NN) )            CRSX0114
      TMAX = DMIN1( E(IG,NN) - EBI(IH+1), GAMMA(1) * E(IG,NN) )         CRSX0115
      SCAT = 0D0                                                        CRSX0116
      IF( TMIN .GE. TMAX ) GO TO 80                                     CRSX0117
      CALL QUANC8( DSIG, TMIN, TMAX, 0D0, 1D-5, SCAT, ERREST,           CRSX0118
     1                                                  NOFUN, FLAG)    CRSX0119
      IF( FLAG .NE. 0D0 ) STOP 12                                       CRSX0120
      CALL LEGEND( E(IG,NN), TMIN, TMAX, 0)                             CRSX0121
   80 PSIGS(IG,IH,NN) = SCAT                                            CRSX0122
      IF( NT .EQ. 0 ) GO TO 100                                         CRSX0123
C                                                                       CRSX0124
C  CALCULATE POINT-ENERGY SCATTERING CROSS SECTION FOR INTERACTIONS IN  CRSX0125
C  WHICH THE RECOIL TARGET ATOM EMERGES IN GROUP IH:                    CRSX0126
      DO 90 IH = 1, NT                                                  CRSX0127
      IHI = NI + IH                                                     CRSX0128
      TMIN = DMAX1( EBT(IH+1), TLOW(1) * E(IG,NN) )                     CRSX0129
      TMAX = DMIN1( EBT(IH), GAMMA(1) * E(IG,NN) )                      CRSX0130
      SCAT = 0D0                                                        CRSX0131
      IF( TMIN .GE. TMAX ) GO TO 90                                     CRSX0132
```

Appendix A (contd.)

```
      CALL QUANC8( DSIG, TMIN, TMAX, 0D0, 1D-5, SCAT, ERREST,          CRSX0133
     1                                                 NOFUN, FLAG)   CRSX0134
      IF( FLAG .NE. 0D0 ) STOP 13                                      CRSX0135
      CALL LEGEND( E(IG,NN), TMIN, TMAX, 1)                            CRSX0136
   90 PSIGS(IG,IHH,NN) = SCAT                                          CRSX0137
  100 CONTINUE                                                         CRSX0138
  110 CONTINUE                                                         CRSX0139
      IF( NT .EQ. 0 ) GO TO 160                                        CRSX0140
C                                                                      CRSX0141
C  END OF ION POINT-ENERGY CROSS SECTION EVALUATION.                   CRSX0142
C                                                                      CRSX0143
C**********BEGIN TARGET ATOM POINT-ENERGY CROSS SECTION EVALUATION*****CRSX0144
C                                                                      CRSX0145
C  SET ION FLAG AND START LOOP OVER ALL TARGET ATOM GROUPS:            CRSX0146
      ION = 2                                                          CRSX0147
      DO 150 IG = 1, NT                                                CRSX0148
C                                                                      CRSX0149
C  CALCULATE STOPPING CROSS SECTION AT LOWER ENERGY GROUP BOUNDARIES:  CRSX0150
      IGG = NI + IG                                                    CRSX0151
      CALL TSIGMA( EBT(IG+1) )                                         CRSX0152
      TMIN = 0D0                                                       CRSX0153
      TMAX = TLOW(2) * EBT(IG+1)                                       CRSX0154
      CALL QUANC8( TDSIG, TMIN, TMAX, 0D0, 1D-4, ELAS, ERREST,         CRSX0155
     1                                                 NOFUN, FLAG)   CRSX0156
      IF( FLAG .NE. 0D0 ) STOP 14                                      CRSX0157
      S(IGG) = ELAS + ESTOP(2) * DSQRT( EBT(IG+1) )                    CRSX0158
C                                                                      CRSX0159
C  EVALUATE TOTAL POINT-ENERGY CROSS SECTION FOR DISCRETE INTERACTIONS: CRSX0160
      DO 140 NN = 1, INP                                               CRSX0161
      CALL TSIGMA( E(IGG,NN) )                                         CRSX0162
      TMIN = TLOW(2) * E(IGG,NN)                                       CRSX0163
      TMAX = E(IGG,NN)                                                 CRSX0164
      CALL QUANC8( DSIG, TMIN, TMAX, 0D0, 1D-5, TOT, ERREST,           CRSX0165
     1                                                 NOFUN, FLAG)   CRSX0166
      IF( FLAG .NE. 0D0 ) STOP 15                                      CRSX0167
      PSIGT(IGG,NN) = TOT                                              CRSX0168
C                                                                      CRSX0169
C  CALCULATE POINT-ENERGY SCATTERING CROSS SECTION FOR INTERACTIONS IN CRSX0170
C  WHICH THE INCIDENT TARGET ATOM EMERGES IN GROUP IH:                 CRSX0171
      DO 130 IH = IG, NT                                               CRSX0172
      IHH = IH + NI                                                    CRSX0173
      TMIN = DMAX1( E(IGG,NN) - EBT(IH), TLOW(2) * E(IGG,NN) )         CRSX0174
      TMAX = DMIN1( E(IGG,NN) - EBT(IH+1), E(IGG,NN) )                 CRSX0175
      SCAT = 0D0                                                       CRSX0176
      IF( TMIN .GE. TMAX ) GO TO 120                                   CRSX0177
      CALL QUANC8( DSIG, TMIN, TMAX, 0D0, 1D-5, SCAT, ERREST,          CRSX0178
     1                                                 NOFUN, FLAG)   CRSX0179
      IF( FLAG .NE. 0D0 ) STOP 16                                      CRSX0180
      CALL LEGEND( E(IGG,NN), TMIN, TMAX, 0)                           CRSX0181
  120 PSIGS(IGG,IHH,NN) = SCAT                                         CRSX0182
      PSIGS(IGG,IH,NN) = SCAT                                          CRSX0183
C                                                                      CRSX0184
```

Appendix A (contd.)

```
C  CALCULATE POINT-ENERGY SCATTERING CROSS SECTION FOR INTERACTIONS IN  CRSXC185
C  WHICH THE RECOIL TARGET ATOM EMERGES IN GROUP IH:                    CRSXC186
      TMIN = DMAX1( EBT(IH+1), TLOW(2) * E(IGG,NN) )                    CRSXC187
      TMAX = DMIN1( EBT(IH), E(IGG,NN) )                                CRSXC188
      SCAT = 0D0                                                        CRSXC189
      IF( TMIN .GE. TMAX ) GO TO 130                                    CRSXC190
      CALL GUANC8(DSIG, TMIN, TMAX, 0D0, 1D-5, SCAT, ERREST,            CRSXC191
     1                                                  NOFUN, FLAG)    CRSXC192
      IF( FLAG .NE. 0D0 ) STOP 17                                       CRSXC193
      CALL LEGEND( E(IGG,NN), TMIN, TMAX, 1)                            CRSXC194
  130 PSIGS(IGG,IHH,NN) = PSIGS(IGG,IHH,NN) + SCAT                      CRSXC195
  140 CONTINUE                                                          CRSXC196
  150 CONTINUE                                                          CRSXC197
C                                                                       CRSXC198
C  END OF TARGET ATOM POINT-ENERGY CROSS SECTION EVALUATION.            CRSXC199
C                                                                       CRSXC200
C*******START FLUX ITERATION AND MULTIGROUP (P0) CALCULATIONS**********CRSXC201
  160 ITER = 0                                                          CRSXC202
      DAMP = 0.5D0                                                      CRSXC203
C                                                                       CRSX0204
C  EVALUATE XI, THE EXPONENT OF THE WEIGHTING FUNCTION:                 CRSX0205
  170 XN = PHIO(2) * (EBI(1) - EBI(2)) / (PHIO(1) * (EBI(2) - EBI(3)))  CRSX0206
      XD = ( EBI(1) + EBI(2) ) / ( EBI(2) + EBI(3) )                    CRSX0207
      QQ = DLOG( XN ) / DLOG( XD )                                      CRSX0208
      IF( ITER .EQ. 0 ) XI(1) = QQ                                      CRSX0209
      XI(1) = ( 1D0 - DAMP ) * QQ + DAMP * XI(1)                        CRSX0210
      DO 180 IG = 2, NI                                                 CRSX0211
      XN = XI(IG-1) *                                                   CRSX0212
     1          DLOG( ( EBI(IG-1) + EBI(IG) ) / ( EBI(IG) + EBI(IG) ) ) CRSX0213
     2                - DLOG( PHIO(IG) * ( EBI(IG-1) - EBI(IG) ) /    CRSX0214
     3                ( PHIO(IG-1) * (EBI(IG) - EBI(IG+1) ) ) )       CRSX0215
      XD = DLOG( ( EBI(IG) + EBI(IG+1) ) / ( EBI(IG) + EBI(IG) ) )      CRSX0216
      QQ = XN / XD                                                      CRSX0217
      IF( ITER .EQ. 0 ) XI(IG) = QQ                                     CRSX0218
  180 XI(IG) = ( 1D0 - DAMP ) * QQ + DAMP * XI(IG)                      CRSX0219
      IF( NT .EQ. 0 ) GO TO 200                                         CRSX0220
      XN = PHIO(NIP2) * ( EBT(1) - EBT(2) ) /                           CRSX0221
     1                        ( PHIO(NIP1) * ( EBT(2) - EBT(3) ) )      CRSX0222
      XD = ( EBT(1) + EBT(2) ) / ( EBT(2) + EBT(3) )                    CRSX0223
      QQ = DLOG( XN ) / DLOG( XD )                                      CRSX0224
      IF( ITER .EQ. 0 ) XI(NIP1) = QQ                                   CRSX0225
      XI(NIP1) = ( 1D0 - DAMP ) * QQ + DAMP * XI(NIP1)                  CRSX0226
      DO 190 IG = NIP2, NTOT                                            CRSX0227
      IH = IG - NI                                                      CRSX0228
      XN = XI(IG-1) *                                                   CRSX0229
     1          DLOG( ( EBT(IH-1) + EBT(IH) ) / ( EBT(IH) + EBT(IH) ) ) CRSX0230
     2          - DLOG( PHIO(IG) * ( EBT(IH-1) - EBT(IH) ) /            CRSX0231
     3                ( PHIO(IG-1) * (EBT(IH) - EBT(IH+1) ) ) )       CRSX0232
      XD = DLOG( ( EBT(IH) + EBT(IH+1) ) / ( EBT(IH) + EBT(IH) ) )      CRSX0233
      QQ = XN / XD                                                      CRSX0234
      IF( ITER .EQ. 0 ) XI(IG) = QQ                                     CRSX0235
  190 XI(IG) = ( 1D0 - DAMP ) * QQ + DAMP * XI(IG)                      CRSX0236
```

Appendix A (contd.)

```
C                                                                       CRSX0237
C  CALCULATE DEN(G), THE INTEGRAL OF THE WEIGHT FUNCTION:              CRSX0238
  200 DO 210 IG = 1, NI                                                 CRSX0239
      XI1 = 1D0 - XI(IG)                                                CRSX0240
  210 DEN(IG) = ( EBI(IG) ** XI1 - EBI(IG+1) ** XI1 ) / XI1             CRSX0241
      DO 220 IG = 1, NT                                                 CRSX0242
      XI1 = 1D0 - XI(IG+NI)                                             CRSX0243
  220 DEN(IG+NI) = ( EBT(IG) ** XI1 - EBT(IG+1) ** XI1 ) / XI1          CRSX0244
C                                                                       CRSX0245
C  CALCULATE THE WEIGHTS FOR THE INTEGRATION OVER THE ENERGY GROUPS:    CRSX0246
      DO 240 IG = 1, NI                                                 CRSX0247
      DO 230 NN = 1, INP                                                CRSX0248
      A = 0.5D0 * ( EBI(IG) - EBI(IG+1) )                               CRSX0249
  230 WTS(IG,NN) = A * WT1(NN) * E(IG,NN) ** ( - XI(IG) ) / DEN(IG)     CRSX0250
  240 CONTINUE                                                          CRSX0251
      DO 260 IG = 1, NT                                                 CRSX0252
      DO 250 NN = 1, INP                                                CRSX0253
      A = 0.5D0 * ( EBT(IC) - EBT(IG+1) )                               CRSX0254
  250 WTS(NI+IG,NN) = A * WT1(NN) * E(NI+IG,NN) ** ( - XI(NI+IG) )      CRSX0255
     1                                          / DEN(NI+IG)           CRSX0256
  260 CONTINUE                                                          CRSX0257
C                                                                       CRSX0258
C  CALCULATE THE TOTAL MULTIGROUP CROSS SECTION:                        CRSX0259
      DO 300 IG = 1, NTOT                                               CRSX0260
      SIGT(IG) = 0D0                                                    CRSX0261
      DO 270 NN = 1, INP                                                CRSX0262
  270 SIGT(IG) = SIGT(IG) + DENSE * WTS(IG,NN) * PSIGT(IG,NN)           CRSX0263
C                                                                       CRSX0264
C  CALCULATE THE CONTINUOUS SLOWING DOWN CROSS SECTION:                 CRSX0265
      IF( IG .LE. NI ) SIGSD(IG) =                                      CRSX0266
     1          DENSE * S(IG) * EBI(IG+1) ** ( - XI(IG) ) / DEN(IG)    CRSX0267
      IF( IG .GT. NI ) SIGSD(IG) =                                      CRSX0268
     1          DENSE * S(IG) * EBT(IG-NI+1) ** ( - XI(IG) ) / DEN(IG) CRSX0269
C                                                                       CRSX0270
C  MODIFY THE TOTAL CROSS SECTION FOR 'ANISN':                          CRSX0271
      SIGT(IG) = SIGT(IG) + SIGSD(IG)                                   CRSX0272
C                                                                       CRSX0273
C  CALCULATE THE MULTIGROUP SCATTERING MATRIX, P0 COEFFICIENT:          CRSX0274
      DO 290 IH = IG, NTOT                                              CRSX0275
      SIGS(IG,IH,1) = 0D0                                               CRSX0276
      DO 280 NN = 1, INP                                                CRSX0277
  280 SIGS(IG,IH,1) = SIGS(IG,IH,1) +                                   CRSX0278
     1                            DENSE * WTS(IG,NN) * PSIGS(IG,IH,NN) CRSX0279
  290 CONTINUE                                                          CRSX0280
  300 CONTINUE                                                          CRSX0281
C                                                                       CRSX0282
C  PREVENT IONS FROM ENTERING THE TARGET ATOM GROUPS:                   CRSX0283
      SIGSD(NI) = 0D0                                                   CRSX0284
C                                                                       CRSX0285
C  UPDATE THE FLUXES USING THE NEW MULTIGROUP PARAMETERS:               CRSX0286
      PHI(1) = 1D0 / ( SIGT(1) - SIGS(1,1,1) )                          CRSX0287
      DO 320 IG = 2, NTOT                                               CRSX0288
      SUM = SIGSD(IG-1) * PHI(IG-1)                                     CRSX0289
      IGM1 = IG - 1                                                     CRSX0290
      DO 310 IH = 1, IGM1                                               CRSX0291
```

Appendix A (contd.)

```
  310 SUM = SUM + SIGS(IH,IG,1) * PHI(IH)                                CRSX0292
  320 PHI(IG) = SUM / ( SIGT(IG) - SIGS(IG,IG,1) )                       CRSX0293
      ITER = ITER + 1                                                    CRSX0294
C                                                                        CRSX0295
C  CHECK FOR CONVERGENCE OF THE FLUXES:                                  CRSX0296
      IFLG = 0                                                           CRSX0297
      DO 330 IG = 1, NTOT                                                CRSX0298
      IF( DABS( PHI(IG) - PHIO(IG) ) / PHIO(IG) .GT. 1D-2 ) IFLG = 1     CRSX0299
  330 PHIO(IG) = PHI(IG)                                                 CRSX0300
C                                                                        CRSX0301
C  REPORT FLUXES, IF REQUESTED:                                          CRSX0302
      IF( NEWS .LE. 0 .AND. IFLG .NE. 0 ) GO TO 170                      CRSX0303
      PRINT 340, ITER, ( (IG, PHI(IG)), IG = 1, NTOT)                    CRSX0304
  340 FORMAT(//5X,'FLUXES AFTER ITERATION',I4//8X,'IG',6X,'FLUX',13X,'IGCRSX0305
     1',6X,'FLUX',13X,'IG',6X,'FLUX',13X,'IG',6X,'FLUX',13X,'IG',6X,'FLUCRSX0306
     2X'/(I10,G15.5,I10,G15.5,I10,G15.5,I10,G15.5,I10,G15.5))            CRSX0307
      IF( IFLG .NE. 0 ) GO TO 170                                        CRSX0308
C                                                                        CRSX0309
C  END OF FLUX ITERATION AND MULTIGROUP (P0) CALCULATIONS:               CRSX0310
C                                                                        CRSX0311
C**********CALCULATE HIGHER ORDER LEGENDRE COEFFICIENTS*****************CRSX0312
C                                                                        CRSX0313
C  SAVE THE P0 COEFFICIENT AND CLEAR THE CROSS SECTION MATRIX:           CRSX0314
      DO 370 IG = 1, NTOT                                                CRSX0315
      DO 360 IH = IG, NTOT                                               CRSX0316
      PNORM(IG,IH) = SIGS(IG,IH,1)                                       CRSX0317
      DO 350 L = 1, NCOEF                                                CRSX0318
  350 SIGS(IG,IH,L) = 0D0                                                CRSX0319
  360 CONTINUE                                                           CRSX0320
  370 CONTINUE                                                           CRSX0321
C                                                                        CRSX0322
C  RETRIEVE THE POINT-ENERGY COEFFICIENTS FROM DATA SET REFERENCE        CRSX0323
C  NUMBER 4, WHERE THEY WERE SAVED BY 'LEGEND':                          CRSX0324
      REWIND 4                                                           CRSX0325
C                                                                        CRSX0326
C  ION CALCULATION:                                                      CRSX0327
      DO 430 IG = 1, NI                                                  CRSX0328
      DO 420 NN = 1, INP                                                 CRSX0329
      DO 390 IH = IG, NI                                                 CRSX0330
      IF( PSIGS(IG,IH,NN ) .LE. 0D0 ) GO TO 390                          CRSX0331
      READ(4) (COEF(L), L = 1, NCOEF)                                    CRSX0332
C                                                                        CRSX0333
C  INTEGRATE NUMERICALLY OVER THE ENERGY GROUP TO OBTAIN SIGS(IG,IH,L):  CRSX0334
      DO 380 L = 1, NCOEF                                                CRSX0335
  380 SIGS(IG,IH,L) = SIGS(IG,IH,L) + DENSE * WTS(IG,NN) * COEF(L)       CRSX0336
  390 CONTINUE                                                           CRSX0337
      DO 410 IH = 1, NT                                                  CRSX0338
      IHH = NI + IH                                                      CRSX0339
      IF( PSIGS(IG,IHH,NN) .LE. 0D0 ) GO TO 410                          CRSX0340
      READ(4) (COEF(L), L = 1, NCOEF)                                    CRSX0341
      DO 400 L = 1, NCOEF                                                CRSX0342
```

Appendix A (contd.)

```
  400 SIGS(IG,IHH,L) = SIGS(IC,IHH,L) + DENSE * WTS(IG,NN) * COEF(L)     CRSXC343
  410 CONTINUE                                                          CRSXC344
  420 CONTINUE                                                          CRSXC345
  430 CONTINUE                                                          CRSXC346
C                                                                       CRSXC347
C  TARGET ATOM CALCULATION:                                             CRSXC348
      DO 490 IG = 1, NT                                                 CRSXC349
      IGG = NI + IG                                                     CRSXC350
      DO 480 NN = 1, INP                                                CRSXC351
      DO 470 IH = IG, NT                                                CRSXC352
      IHH = IH + NI                                                     CRSXC353
      IF( PSIGS(IGG,IH,NN) .LE. 0D0 ) GO TO 450                         CRSXC354
      READ(4) (COEF(L), L = 1, NCOEF)                                   CRSXC355
      DO 440 L = 1, NCOEF                                               CRSXC356
  440 SIGS(IGG,IHH,L) = SIGS(IGG,IHH,L) + DENSE * WTS(IGG,NN) * COEF(L) CRSXC357
  450 IF( PSIGS(IGG,IHH,NN) .LE. PSIGS(IGG,IH,NN) ) GO TO 470           CRSXC358
      READ(4) (COEF(L), L = 1, NCOEF)                                   CRSXC359
      DO 460 L = 1, NCOEF                                               CRSXC360
  460 SIGS(IGG,IHH,L) = SIGS(IGG,IHH,L) + DENSE * WTS(IGG,NN) * COEF(L) CRSXC361
  470 CONTINUE                                                          CRSXC362
  480 CONTINUE                                                          CRSXC363
  490 CONTINUE                                                          CRSXC364
C                                                                       CRSXC365
C  RENORMALIZE USING THE HIGHER PRECISION PO COEFFICIENT:               CRSXC366
      DO 520 IG = 1, NTOT                                               CRSXC367
      DO 510 IH = IG, NTOT                                              CRSXC368
      IF( PNORM(IG,IH) .EQ. 0D0 ) GO TO 510                             CRSXC369
      PNORM(IG,IH) = PNORM(IG,IH) / SIGS(IG,IH,1)                       CRSXC370
      DO 500 L = 1, NCOEF                                               CRSXC371
      SIGS(IG,IH,L) = SIGS(IG,IH,L) * PNORM(IG,IH)                      CRSXC372
  500 CONTINUE                                                          CRSXC373
  510 CONTINUE                                                          CRSXC374
  520 CONTINUE                                                          CRSXC375
C                                                                       CRSXC376
C  NORMAL TERMINATION OF THE PROCEDURE:  RETURN TO CALLING PROGRAM.     CRSXC377
      RETURN                                                            CRSXC378
      END                                                               CRSXC379
C                                                                       TSIG0001
      SUBROUTINE TSIGMA(EP)                                             TSIG0002
C  THIS PROCEDURE SETS UP A TABLE OF THE DIFFERENTIAL SCATTERING CROSS   TSIG0003
C  SECTION, 'DSIG', AS A FUNCTION OF 'T', THE ENERGY TRANSFERRED TO THE TSIG0004
C  RECOILING TARGET ATOM.  'EP' IS THE ENERGY OF THE INCIDENT PARTICLE.  TSIG0005
C                                                                       TSIG0006
C  THE TABLE IS BASED ON THE APPROXIMATION:                             TSIG0007
C                                                                       TSIG0008
C       DSIG(T) / DSIG(TM) = (TM / T) ** N                              TSIG0009
C                                                                       TSIG0010
C  WHERE 'TM' IS THE MAXIMUM POSSIBLE TRANSFERRED ENERGY.  THE TABLES   TSIG0011
C  CONTAIN VALUES OF LN( TM / T ) AND OF N.  THE VALUES OF 'DSIG' ARE   TSIG0012
C  RETRIEVED BY THE FUNCTION DSIG(T).  THE TABLES ARE PASSED IN THE     TSIG0013
C  COMMON BLOCK 'TABL'.                                                 TSIG0014
```

Appendix A (contd.)

```
C                                                                       TSIG0015
C  VARIABLE SPECIFICATIONS:                                            TSIG0016
      IMPLICIT REAL*8 (A - H, O - Z)                                    TSIG0017
      REAL*8 DER(2)                                                     TSIG0018
      COMMON/PLAT/NI,NT,NCOEF,ION,DENSE,EI,UO,APQ(2),QGAM(2),RGAM(2),   TSIG0019
     1           SGAM(2),GAMMA(2),TLOW(2),TLOG(2),APAR(2),BPAR(2),      TSIG0020
     2           CPAR(2),ESTOP(2),NEWS,NTOT                             TSIG0021
      COMMON/WATE/U(4),W(4),AB(10),WT1(10),INP,MEHL                     TSIG0022
      COMMON/TABL/TMX,SIGO,KM,KQ,TE(200),SIG(200)                       TSIG0023
      DATA C1,C2,C3/1.4421618D0,6.19086846881D-2,-6.39879712938D-2/     TSIG0024
C                                                                       TSIG0025
      QX = APQ(ION) * EP                                                TSIG0026
      TMX = RGAM(ION) / QX                                              TSIG0027
C                                                                       TSIG0028
C  ESTIMATE THE SPACING IN THE IMPACT PARAMETER NEEDED TO CONSTRUCT     TSIG0029
C  TABLES WITH SUITABLE SPACINGS IN T AND SIGMA FOR INTERPOLATION.      TSIG0030
      XX = ( 0.43429448190325183D0 * DLOG( QX * CPAR(ION) ) ) + 6D0     TSIG0031
      DELS = BPAR(ION) * DEXP( 2.3025850929940457D0 *                   TSIG0032
     1                         ( C1 + XX * ( C2 + C3 * XX ) ) )         TSIG0033
C                                                                       TSIG0034
C  EVALUATE CROSS SECTION IN HEAD-ON COLLISION:                         TSIG0035
      L = 0                                                             TSIG0036
      K = 0                                                             TSIG0037
      SONE = 0D0                                                        TSIG0038
      SAA = 1D-10                                                       TSIG0039
      ASSIGN 120 TO MARK                                                TSIG0040
      GO TO 40                                                          TSIG0041
C                                                                       TSIG0042
C  STEP TO THE NEXT IMPACT PARAMETER VALUE:                             TSIG0043
   20 K = K + 1                                                         TSIG0044
      IF( K .GT. 200 ) STOP 6                                           TSIG0045
      L = L + 1                                                         TSIG0046
      IF( K .NE. 1 ) GO TO 30                                           TSIG0047
      SAA = 1D-5                                                        TSIG0048
      GO TO 40                                                          TSIG0049
   30 SAA = SONE + (L - 1) * DELS                                       TSIG0050
C                                                                       TSIG0051
C  EVALUATION OF THE APSIS OF THE COLLISION, R.                         TSIG0052
C  INITIAL ESTIMATE OF THE APSIS:                                       TSIG0053
   40 QA = CPAR(ION) * QX                                               TSIG0054
      QB = DLOG( QA )                                                   TSIG0055
      QA = DEXP((6.20563D-3 * QB + 3.15458D-1) * QB - 4.54369D-1) + QA  TSIG0056
      R = BPAR(ION) / QA                                                TSIG0057
      IF( SAA .GT. R ) R = SAA                                          TSIG0058
C                                                                       TSIG0059
C  REFINE THE APSIS BY QUADRATIC EXTRAPOLATION:                         TSIG0060
   50 QQ = R - POTENZ(R,DER) / QX                                       TSIG0061
      QA = DER(1) / QX                                                  TSIG0062
      QB = QQ + QQ - QA                                                 TSIG0063
      IF( QB .GT. 0D0 ) GO TO 70                                        TSIG0064
   60 R = R + SAA                                                       TSIG0065
      GO TO 50                                                          TSIG0066
```

Appendix A (contd.)

```
   70 QA = QQ - QA - QA - DER(2) / (QX + QX)                          TSIG0067
      QQ = R * QQ - ( SAA * SAA )                                     TSIG0068
      QA = QQ / ( QB - QA * QQ / ( QB * R ) )                         TSIG0069
      IF( QA .GT. R ) GO TO 60                                        TSIG0070
      R = R - QA                                                      TSIG0071
      IF( DABS(QA) .GT. 1D-3 * R ) GO TO 50                           TSIG0072
C                                                                     TSIG0073
C  CONTINUE REFINEMENT BY FAST LINEAR EXTRAPOLATION:                  TSIG0074
   80 QA = (R * (R - POTENZ(R,DER) / QX) - ( SAA * SAA ) ) / QB       TSIG0075
      R = R - QA                                                      TSIG0076
      IF( DABS(QA) .GT. 1D-8 * R ) GO TO 80                           TSIG0077
C                                                                     TSIG0078
C  THE BARYCENTRIC (CM) SYSTEM SCATTERING INTEGRALS:                  TSIG0079
      SUM = 0D0                                                       TSIG0080
      DTHDS = 0D0                                                     TSIG0081
      IF( R .LE. SAA ) GO TO 90                                       TSIG0082
      GO TO 100                                                       TSIG0083
   90 SA = POTENZ(SAA,DER) / (SAA * QX)                               TSIG0084
      SB = DSQRT( 1D0 - SA )                                          TSIG0085
      R = SAA / SB                                                    TSIG0086
  100 QZ = R / QX                                                     TSIG0087
      PB = POTENZ(R,DER)                                              TSIG0088
      PA = DER(1)                                                     TSIG0089
      PB = DER(1) / R                                                 TSIG0090
C                                                                     TSIG0091
C  GAUSS-MEHLER QUADRATURE OF THE SCATTERING INTEGRALS:               TSIG0092
      DO 110 J = 1, MEHL                                              TSIG0093
      QA = R / U(J)                                                   TSIG0094
      QQ = SAA * U(J)                                                 TSIG0095
      QB = QZ * U(J)                                                  TSIG0096
      QC = POTENZ(QA,DER)                                             TSIG0097
      QD = 1D-70 / QB                                                 TSIG0098
      IF( QD .GT. QC ) QC = 0D0                                       TSIG0099
      QB = QB * QC                                                    TSIG0100
      QQ = (R + QQ) * (R - QQ) - QB                                   TSIG0101
      Q = W(J) / DSQRT( QQ )                                          TSIG0102
      SUM = SUM + Q                                                   TSIG0103
  110 DTHDS = DTHDS + Q * ( PB * U(J) * U(J) - DER(1) / QA ) / QQ     TSIG0104
      PB = SAA * SUM                                                  TSIG0105
      RATIO = ( DSIN( 1.5707963267948966D0 - PB ) ) ** 2              TSIG0106
C                                                                     TSIG0107
C  CALCULATION OF THE DIFFERENTIAL CROSS SECTION:                     TSIG0108
      PA = 1D0 / (1D0 - 0.5D0 * QZ * PA / (SAA * SAA))                TSIG0109
      DTHDS = DABS( QZ * DTHDS * R * PA + (PA - 1D0) * (SUM + SUM) )  TSIG0110
      PA = 1D-10 * SUM                                                TSIG0111
      IF( PB .LT. PA )  B = PA                                        TSIG0112
      IF( RATIO .LT. 1D-10 ) PB = 1D-10                               TSIG0113
      DTHDS = 12.566370614390D0 * TMX * SAA / ( DTHDS * DSIN( PB + PB ) )TSIG0114
      GO TO MARK, ( 120, 130, 140)                                    TSIG0115
  120 SIGO = DTHDS                                                    TSIG0116
      ASSIGN 130 TO MARK                                              TSIG0117
      GO TO 20                                                        TSIG0118
```

Appendix A (contd.)

```
  130 TE(K) = - DLOG( RATIO )                                          TSIG0119
      SIG(K) = DLOG( DTHDS / SIGO ) / TE(K)                            TSIG0120
      IF( L .LT. 4 ) GO TO 20                                          TSIG0121
      SFOUR = SAA                                                      TSIG0122
      SAA = SONE + 1.5D0 * DELS                                        TSIG0123
      ASSIGN 140 TO MARK                                               TSIG0124
      GO TO 40                                                         TSIG0125
C                                                                      TSIG0126
C  TEST THE CURRENT IMPACT PARAMETER SPACING:                          TSIG0127
  140 ASSIGN 130 TO MARK                                               TSIG0128
      QQ = - DLOG( RATIO )                                             TSIG0129
      PA = SIG(K-3) * (QQ - TE(K-2)) * (QQ - TE(K-1)) * (QQ - TE(K)) /  TSIG0130
     1  ((TE(K-3) - TE(K-2)) * (TE(K-3) - TE(K-1)) * (TE(K-3) - TE(K)))  TSIG0131
      PA = PA +                                                        TSIG0132
     1     SIG(K-2) * (QQ - TE(K-3)) * (QQ - TE(K-1)) * (QQ - TE(K)) /  TSIG0133
     2  ((TE(K-2) - TE(K-3)) * (TE(K-2) - TE(K-1)) * (TE(K-2) - TE(K)))  TSIG0134
      PA = PA +                                                        TSIG0135
     1     SIG(K-1) * (QQ - TE(K-3)) * (QQ - TE(K-2)) * (QQ - TE(K)) /  TSIG0136
     2  ((TE(K-1) - TE(K-3)) * (TE(K-1) - TE(K-2)) * (TE(K-1) - TE(K)))  TSIG0137
      PA = PA +                                                        TSIG0138
     1     SIG(K) * (QQ - TE(K-3)) * (QQ - TE(K-2)) * (QQ - TE(K-1)) /  TSIG0139
     2  ((TE(K) - TE(K-3)) * (TE(K) - TE(K-2)) * (TE(K) - TE(K-1)))      TSIG0140
      PA = DABS( SIGO * DEXP( PA * QQ ) / DTHDS - 1D0 )                TSIG0141
      IF( PA .GT. 1D-3 ) GO TO 150                                     TSIG0142
      IF( PA .GT. 1D-4 ) GO TO 160                                     TSIG0143
C                                                                      TSIG0144
C  INCREASE SPACING OF IMPACT PARAMETER:                               TSIG0145
      DELS = 1.6D0 * DELS                                              TSIG0146
      L = 0                                                            TSIG0147
      K = K - 4                                                        TSIG0148
      GO TO 20                                                         TSIG0149
C                                                                      TSIG0150
C  DECREASE IMPACT PARAMETER SPACING:                                  TSIG0151
  150 DELS = 0.9D0 * DELS                                              TSIG0152
      L = 0                                                            TSIG0153
      K = K - 4                                                        TSIG0154
      GO TO 20                                                         TSIG0155
C                                                                      TSIG0156
C  CONTINUE WITH CURRENT IMPACT PARAMETER SPACING:                     TSIG0157
  160 IF( TE(K) .GT. TLOG(ION) ) GO TO 170                             TSIG0158
      SONE = SFOUR + DELS                                              TSIG0159
      L = 0                                                            TSIG0160
      GO TO 20                                                         TSIG0161
C                                                                      TSIG0162
C  PRINT TABLE IF REQUESTED AND RETURN TO CALLING PROCEDURE:           TSIG0163
  170 KM = K                                                           TSIG0164
      KQ = K - 1                                                       TSIG0165
      IF( NEWS .GT. 2 ) PRINT 180, ION, EP, TLOG(ION), KM, TMX, SIGO,  TSIG0166
     1                             (J, TE(J), SIG(J), J = 1, KM)       TSIG0167
      RETURN                                                           TSIG0168
C                                                                      TSIG0169
```

Appendix A (contd.)

```
C  OUTPUT FORMAT:                                                        TSIG0170
  180 FORMAT(///30X,'INTERPOLATION TABLE FOR THE DIFFERENTIAL ELASTIC SCTSIG0171
     1ATTERING CROSS SECTION'//11X,'ION',I2,5X,'EP',G15.8,5X,'TLOG',G15.TSIG0172
     28,5X,'KM',I5,5X,'TMX',G15.8,5X,'SIGO',G15.8//8X,'K',5X,'TE(K)',11XTSIG0173
     3,'SIG(K)',14X,'K',5X,'TE(K)',11X,'SIG(K)',14X,'K',5X,'TE(K)',11X,'TSIG0174
     4SIG(K)'/(I10,2G16.8,I10,2G16.8,I10,2G16.8))                        TSIG0175
      END                                                                TSIG0176
C                                                                        DSIG0001
      FUNCTION DSIG( T )                                                 DSIG0002
C  INTERPOLATES THE TABLES CONSTRUCTED BY TSIGMA TO PROVIDE VALUES OF    DSIG0003
C  THE DIFFERENTIAL CROSS SECTION FOR TRANSFERRING ENERGY T.             DSIG0004
      IMPLICIT REAL*8 (A - H, O - Z)                                     DSIG0005
      COMMON/TABL/TMX,SIGO,KM,KQ,TE(200),SIG(200)                        DSIG0006
C                                                                        DSIG0007
      TQ = T                                                             DSIG0008
      ASSIGN 90 TO JTD                                                   DSIG0009
      GO TO 10                                                           DSIG0010
C                                                                        DSIG0011
      ENTRY TDSIG( T )                                                   DSIG0012
C  SUPPLIES THE VALUE OF T * DSIG INSTEAD OF DSIG.                       DSIG0013
      TQ = T                                                             DSIG0014
      IF( TQ .LT. 1D-4 ) TQ = 1D-4                                       DSIG0015
      ASSIGN 80 TO JTD                                                   DSIG0016
C                                                                        DSIG0017
   10 QQ = - DLOG( TQ * TMX )                                            DSIG0018
      IF( QQ .LT. 0D0 ) QQ = 0D0                                         DSIG0019
      IF( KM .GT. 4 ) GO TO 20                                           DSIG0020
      K = 3                                                              DSIG0021
      GO TO 60                                                           DSIG0022
   20 KA = 2                                                             DSIG0023
      KZ = KQ                                                            DSIG0024
   30 K = ( KA + KZ ) / 2                                                DSIG0025
      IF( QQ .GT. TE(K) ) GO TO 40                                       DSIG0026
      IF( K - KA .LE. 1 ) GO TO 60                                       DSIG0027
      KZ = K                                                             DSIG0028
      GO TO 30                                                           DSIG0029
   40 IF( KZ - K .LE. 1 ) GO TO 50                                       DSIG0030
      KA = K                                                             DSIG0031
      GO TO 30                                                           DSIG0032
   50 K = KZ                                                             DSIG0033
   60 YF = SIG(K-2) * (QQ - TE(K-1)) * (QQ - TE(K)) * (QQ - TE(K+1)) /   DSIG0034
     1  ((TE(K-2) - TE(K-1)) * (TE(K-2) - TE(K)) * (TE(K-2) - TE(K+1)))  DSIG0035
      YF = YF +                                                          DSIG0036
     1     SIG(K-1) * (QQ - TE(K-2)) * (QQ - TE(K)) * (QQ - TE(K+1)) /   DSIG0037
     2  ((TE(K-1) - TE(K-2)) * (TE(K-1) - TE(K)) * (TE(K-1) - TE(K+1)))   DSIG0038
      YF = YF +                                                          DSIG0039
     1     SIG(K) * (QQ - TE(K-2)) * (QQ - TE(K-1)) * (QQ - TE(K+1)) /   DSIG0040
     2  ((TE(K) - TE(K-2)) * (TE(K) - TE(K-1)) * (TE(K) - TE(K+1)))       DSIG0041
      YF = YF +                                                          DSIG0042
     1     SIG(K+1) * (QQ - TE(K-2)) * (QQ - TE(K-1)) * (QQ - TE(K)) /   DSIG0043
     2  ((TE(K+1) - TE(K-2)) * (TE(K+1) - TE(K-1)) * (TE(K+1) - TE(K)))   DSIG0044
      YF = YF * QQ                                                       DSIG0045
      IF( YF .GT. - 174D0 ) GO TO 70                                     DSIG0046
      DSIG = 0D0                                                         DSIG0047
      GO TO JTD, (80, 90)                                                DSIG0048
```

Appendix A (contd.)

```
   70 IF( YF .GT. 174D0 ) STOP 9                                          DSIG0049
      DSIG = SIGO * DEXP( YF )                                            DSIG0050
      GO TO JTD, (80, 90)                                                 DSIG0051
   80 TDSIG = TQ * DSIG                                                   DSIG0052
   90 RETURN                                                              DSIG0053
      END                                                                 DSIG0054
C                                                                         PTZA0001
      FUNCTION POTENZ( R, DER )                                           PTZA0002
C  VERSION A.  THE MOLIERE APPROXIMATION TO THE THOMAS-FERMI INTER-       PTZA0003
C  ATOMIC POTENTIAL FUNCTION AND ITS DERIVATIVES.  'POTENZ = R * V(R)'.   PTZA0004
C  'DER(1) = R**2 * DV(R)/DR'.  'DER(2) = R**3 * D(DV(R)/LR)/DR'.         PTZA0005
      IMPLICIT REAL*8 (A - H, O - Z)                                      PTZA0006
      REAL*8 DER(2),COEFF(9)                                              PTZA0007
C                                                                         PTZA0008
      COMMON/PDAT/NI,NT,NCOEF,ION,DENSE,RI,U0,APQ(2),QGAM(2),RGAM(2),     PTZA0009
     1            SGAM(2),GAMMA(2),TLOW(2),TLOG(2),APAR(2),BPAR(2),       PTZA0010
     2            CPAR(2),ESTOP(2),NEWS,NTOT                              PTZA0011
C                                                                         PTZA0012
C  VALUES OF NUMERICAL CONSTANTS.  THE VALUES AFTER THE FIRST THREE ARE   PTZA0013
C  11/7, 44/7, 176/7, 2/7, 40/7, 800/7, RESPECTIVELY.                     PTZA0014
      DATA COEFF/1.77D0, 44.25D0, 8.85D0, 1.5714285714285714D0,           PTZA0015
     1           6.2857142857142857D0, 25.142857142857143D0,              PTZA0016
     2           0.2857142857142857D0, 5.7142857142857143D0,              PTZA0017
     3           114.28571428571429D0/                                   PTZA0018
C                                                                         PTZA0019
      FQ = 0.3D0 * R / BPAR(ION)                                          PTZA0020
      IF( FQ .GT. COEFF(1) ) GO TO 20                                     PTZA0021
      F1 = DEXP( - FQ )                                                   PTZA0022
      FEXP = F1                                                           PTZA0023
      GEXP = F1                                                           PTZA0024
      HEXP = F1                                                           PTZA0025
      IF( FQ .GT. COEFF(2) ) GO TO 10                                     PTZA0026
      F1 = ( F1 * F1 ) ** 2                                               PTZA0027
      FEXP = FEXP + COEFF(4) * F1                                         PTZA0028
      GEXP = GEXP + COEFF(5) * F1                                         PTZA0029
      HEXP = HEXP + COEFF(6) * F1                                         PTZA0030
      IF( FQ .GT. COEFF(3) ) GO TO 10                                     PTZA0031
      F1 = F1 * ( F1 * F1 ) ** 2                                          PTZA0032
      FEXP = FEXP + COEFF(7) * F1                                         PTZA0033
      GEXP = GEXP + COEFF(8) * F1                                         PTZA0034
      HEXP = HEXP + COEFF(9) * F1                                         PTZA0035
   10 POTENZ = APAR(ION) * FEXP                                           PTZA0036
      GEXP = FEXP + FQ * GEXP                                             PTZA0037
      DER(1) = - APAR(ION) * GEXP                                         PTZA0038
      DER(2) = APAR(ION) * ( GEXP + GEXP + FQ * FQ * HEXP )               PTZA0039
      RETURN                                                              PTZA0040
   20 POTENZ = 0D0                                                        PTZA0041
      DER(1) = 0D0                                                        PTZA0042
      DER(2) = 0D0                                                        PTZA0043
      RETURN                                                              PTZA0044
      END                                                                 PTZA0045
```

Appendix A (contd.)

```
C                                                                       LGND0001
      SUBROUTINE LEGEND(E, TLO, THI, KOIL)                              LGND0002
C  CALCULATES THE LEGENDRE COEFFICIENTS, COEF, FOR SCATTERINGS WITH AN  LGND0003
C  INCIDENT ENERGY OF E AND RECOIL ENERGIES IN (TLO, THI).              LGND0004
C       IF KOIL = 1, THE PROJECTILE IS THE RECOIL ATOM.                 LGND0005
C       IF KOIL = 0, THE PROJECTILE IS THE INCIDENT PARTICLE.           LGND0006
C                                                                       LGND0007
      IMPLICIT REAL*8 (A-H,O-Z)                                         LGND0008
      DIMENSION P(20),COEF(20)                                          LGND0009
      COMMON/PDAT/MI,MT,NCOEF,ION,DENSE,EI,UO,APQ(2),QGAM(2),RGAM(2),   LGND0010
     1            SGAM(2),GAMMA(2),TLOW(2),TLOG(2),APAR(2),BPAR(2),     LGND0011
     2            CPAR(2),ESTOP(2),NEWS,NTOT                            LGND0012
      COMMON/WATE/U(4),W(4),AB(10),WT1(10),INP,MEHL                     LGND0013
C                                                                       LGND0014
      IF( KOIL .EQ. 1 ) GO TO 10                                        LGND0015
      ASSIGN 50 TO KOILX                                                LGND0016
      GO TO 20                                                          LGND0017
   10 ASSIGN 40 TO KOILX                                                LGND0018
   20 A = 0.5D0 * ( THI - TLO )                                         LGND0019
      B = 0.5D0 * ( THI + TLO )                                         LGND0020
      QA = DSQRT( E )                                                   LGND0021
      DO 30 L = 1, NCOEF                                                LGND0022
   30 COEF(L) = 0D0                                                     LGND0023
C                                                                       LGND0024
C   INTEGRATE FROM TLO TO THI USING INP-POINT GAUSS-LEGENDRE QUADRATURE LGND0025
      DO 90 NN = 1, INP                                                 LGND0026
      T = A * AB(NN) + B                                                LGND0027
      SIG = A * WT1(NN) * DSIG( T )                                     LGND0028
      QB = DSQRT( E - T )                                               LGND0029
C                                                                       LGND0030
C  P(2) IS THE COSINE OF THE ANGLE BETWEEN THE INCIDENT PARTICLE AND    LGND0031
C  THE PROJECTILE UNDER CONSIDERATION.                                  LGND0032
      P(1) = 1D0                                                        LGND0033
      P(2) = ( E - QGAM(ION) * T ) / ( QA * QB )                        LGND0034
      GO TO KOILX, ( 40, 50)                                            LGND0035
   40 P(2) = SGAM(ION) * ( QA - QB * P(2) ) / DSQRT( T )                LGND0036
   50 IF( NCOEF .LT. 3 ) GO TO 70                                       LGND0037
      XMU = P(2)                                                        LGND0038
C                                                                       LGND0039
C   HIGHER ORDER POLYNOMIALS ARE FOUND WITH RECURRENCE RELATION.        LGND0040
      DO 60 LL = 3, NCOEF                                               LGND0041
      EL2 = LL - 2                                                      LGND0042
   60 P(LL) = ( (EL2 + EL2 + 1D0) * XMU * P(LL-1) - EL2 * P(LL-2) ) /   LGND0043
     1                                                  (EL2 + 1D0)     LGND0044
   70 DO 80 LL = 1, NCOEF                                               LGND0045
      EL = LL + LL - 1                                                  LGND0046
   80 COEF(LL) = COEF(LL) + EL * P(LL) * SIG                            LGND0047
   90 CONTINUE                                                          LGND0048
C                                                                       LGND0049
C   TEMPORARILY STORE COEFFICIENTS ON UNIT 4                            LGND0050
      WRITE(4) (COEF(L), L = 1, NCOEF)                                  LGND0051
      RETURN                                                            LGND0052
      END                                                               LGND0053
```

Appendix A (contd.)

```
C                                                                       MEHL0001
      SUBROUTINE MEHLER( N, WO, U, W)                                   MEHL0002
C  CONSTRUCTS TABLES OF ABSCISSAS, U(J), AND WEIGHTS, WO AND W(J), FOR  MEHL0003
C  GAUSS-MEHLER (CHEBYSHEV) QUADRATURE OF INTEGRALS FROM A TO B OF      MEHL0004
C  G(U) * DU.  THE INTEGRAL IS APPROXIMATED BY C * ( SUM FROM J = 1     MEHL0005
C  TO J = N OF D(J) * F(V(J)) ), WHERE C, D(J), V(J), AND F(X) ARE AS   MEHL0006
C  DEFINED BELOW:                                                       MEHL0007
C                                                                       MEHL0008
C     A. N-POINT QUADRATURE OF G(U) = F(U), WHERE F(U) IS ARBITRARY:    MEHL0009
C     C = 2(B - A),  D(J) = W(J) * U(J),  V(J) = A + (B-A) * U(J)**2    MEHL0010
C                                                                       MEHL0011
C     B. N-POINT QUADRATURE OF                                          MEHL0012
C              G(U) = F(U) / SQRT(U * (B + A - U) - A * B)              MEHL0013
C        WHERE F(U) IS ARBITRARY:                                       MEHL0014
C          C = 2 * WO,  D(J) = 1,  V(J) = A + (B - A) * U(J)**2         MEHL0015
C                                                                       MEHL0016
C     C. 2N-POINT QUADRATURE OF G(U) = F(U), WHERE F(U) IS SYMMETRICAL  MEHL0017
C        ABOUT U = A:                                                   MEHL0018
C          C = B - A,  D(J) = W(J),  V(J) = A + (B - A) * U(J)          MEHL0019
C                                                                       MEHL0020
C     D. 2N-POINT QUADRATURE OF                                         MEHL0021
C              G(U) = F(U) / SQRT( (B - U) * (B - 2 * A + U) ),         MEHL0022
C        WHERE F(U) IS SYMMETRICAL ABOUT U = A:                         MEHL0023
C          C = WO,  D(J) = 1,  V(J) = A + (B - A) * U(J)                MEHL0024
C                                                                       MEHL0025
      IMPLICIT REAL*8 (A - H, O - Z)                                    MEHL0026
      REAL*8 U(N),W(N)                                                  MEHL0027
      WO = 1.57079632679489660D0 / N                                    MEHL0028
      Y = DCOS( WO )                                                    MEHL0029
      Z = 5D-1 * Y                                                      MEHL0030
      U(1) = DSQRT( 5D-1 + Z )                                          MEHL0031
      U(N) = DSQRT( 5D-1 - Z )                                          MEHL0032
      Z = U(1) * U(N) + U(1) * U(N)                                     MEHL0033
      W(1) = WO * U(N)                                                  MEHL0034
      W(N) = WO * U(1)                                                  MEHL0035
      IF( N .LE. 1 ) RETURN                                             MEHL0036
      LIM = (N + 1) / 2                                                 MEHL0037
      DO 10 J = 2, LIM                                                  MEHL0038
      U(J) = Y * U(J-1) - Z * U(N-J+2)                                  MEHL0039
      U(N-J+1) = Z * U(J-1) + Y * U(N-J+2)                              MEHL0040
      W(J) = WO * U(N-J+1)                                              MEHL0041
   10 W(N-J+1) = WO * U(J)                                              MEHL0042
      RETURN                                                            MEHL0043
      END                                                               MEHL0044
C                                                                       GAUS0001
C                                                                       GAUS0002
      SUBROUTINE GAUSS(N,T,W,B)                                         GAUS0003
C                                                                       GAUS0004
C                                                                       GAUS0005
```

Appendix A (contd.)

```
C       THIS ROUTINE COMPUTES THE ABSCISSAS T(J) AND WEIGHTS W(J) FOR  GAUS0006
C       AN N-POINT GAUSS-LEGENDRE QUADRATURE.  THESE ARE USED WHEN     GAUS0007
C       ONE WISHES TO APPROXIMATE THE INTEGRAL FROM -1 TO +1 OF F(X)   GAUS0008
C                                                                      GAUS0009
C                              N                                       GAUS0010
C       BY                    SUM W  F(T )                             GAUS0011
C                             J=1 J    J                               GAUS0012
C                                                                      GAUS0013
C       B         REAL SCRATCH ARRAY OF LENGTH N                       GAUS0014
C                                                                      GAUS0015
C     OUTPUT PARAMETERS (BOTH DOUBLE PRECISION ARRAYS OF LENGTH N)     GAUS0016
C                                                                      GAUS0017
C       T         WILL CONTAIN THE DESIRED NODES.                      GAUS0018
C       W         WILL CONTAIN THE DESIRED WEIGHTS W(J).               GAUS0019
C                                                                      GAUS0020
C                                                                      GAUS0021
C     ACCURACY                                                         GAUS0022
C                                                                      GAUS0023
C       THE ROUTINE WAS TESTED UP TO N = 512.  COMPARISON WITH         GAUS0024
C       TABLES IN REF. 3 SHOWED 12 OR MORE SIGNIFICANT DIGITS          GAUS0025
C       OF ACCURACY.                                                   GAUS0026
C                                                                      GAUS0027
C     METHOD                                                           GAUS0028
C                                                                      GAUS0029
C          THE COEFFICIENTS OF THE THREE-TERM RECURRENCE RELATION      GAUS0030
C       FOR THE CORRESPONDING SET OF ORTHOGONAL POLYNOMIALS ARE        GAUS0031
C       USED TO FORM A SYMMETRIC TRIDIAGONAL MATRIX, WHOSE             GAUS0032
C       EIGENVALUES (DETERMINED BY THE IMPLICIT QL-METHOD WITH         GAUS0033
C       SHIFTS) ARE JUST THE DESIRED NODES.  THE FIRST COMPONENTS OF   GAUS0034
C       THE ORTHONORMALIZED EIGENVECTORS, WHEN PROPERLY SCALED,        GAUS0035
C       YIELD THE WEIGHTS.  THIS TECHNIQUE IS MUCH FASTER THAN USING A GAUS0036
C       ROOT-FINDER TO LOCATE THE ZEROES OF THE ORTHOGONAL POLYNOMIAL. GAUS0037
C       FOR FURTHER DETAILS, SEE REF. 1.                               GAUS0038
C                                                                      GAUS0039
C     REFERENCES                                                       GAUS0040
C                                                                      GAUS0041
C       1.  GOLUB, G. H., AND WELSCH, J. H., "CALCULATION OF GAUSSIAN  GAUS0042
C           QUADRATURE RULES," MATHEMATICS OF COMPUTATION 23 (APRIL,   GAUS0043
C           1969), PP. 221-230.                                        GAUS0044
C                                                                      GAUS0045
C       2.  GOLUB, G. H., "SOME MODIFIED MATRIX EIGENVALUE PROBLEMS,"  GAUS0046
C           SIAM REVIEW 15 (APRIL, 1973), PP. 318-334 (SECTION 7).     GAUS0047
C                                                                      GAUS0048
C       3.  STROUD AND SECREST, GAUSSIAN QUADRATURE FORMULAS, PRENTICE-GAUS0049
C           HALL, ENGLEWOOD CLIFFS, N.J., 1966.                        GAUS0050
C                                                                      GAUS0051
      IMPLICIT REAL*8 (A - H, O - Z)                                   GAUS0052
      REAL*8 T(N),W(N),B(N)                                            GAUS0053
      NM1 = N - 1                                                      GAUS0054
      W(1) = 1D0                                                       GAUS0055
      DO 10 I = 1, NM1                                                 GAUS0056
      T(I) = 0D0                                                       GAUS0057
      W(I+1) = 0D0                                                     GAUS0058
      ABI = I                                                          GAUS0059
```

Appendix A (contd.)

```
   10 B(I) = ABI / DSQRT( 4D0  * ABI * ABI - 1D0)                    GAUS0060
      T(N) = 0D0                                                     GAUS0061
      CALL IMTQL2( N, T, B, W, IERR )                                GAUS0062
      IF( IERR .NE. 0 ) STOP 18                                      GAUS0063
      DO 20 I = 1, N                                                 GAUS0064
   20 W(I) = 2D0 * W(I) * W(I)                                       GAUS0065
      RETURN                                                         GAUS0066
      END                                                            GAUS0067
C                                                                    IMTQ0001
      SUBROUTINE IMTQL2( N, D, E, Z, IERR)                           IMTQ0002
C                                                                    IMTQ0003
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE IMTQL2, IMTQ0004
C     NUM. MATH. 12, 377-383(1968) BY MARTIN AND WILKINSON,          IMTQ0005
C     AS MODIFIED IN NUM. MATH. 15, 450(1970) BY DUBRULLE.           IMTQ0006
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 241-248(1971). IMTQ0007
C     THIS IS A MODIFIED VERSION OF THE 'EISPACK' ROUTINE IMTQL2.    IMTQ0008
C                                                                    IMTQ0009
C     THIS SUBROUTINE FINDS THE EIGENVALUES AND FIRST COMPONENTS OF THE IMTQ0010
C     EIGENVECTORS OF A SYMMETRIC TRIDIAGONAL MATRIX BY THE IMPLICIT QL IMTQ0011
C     METHOD.                                                        IMTQ0012
C                                                                    IMTQ0013
C     ON INPUT:                                                      IMTQ0014
C                                                                    IMTQ0015
C        N IS THE ORDER OF THE MATRIX;                               IMTQ0016
C                                                                    IMTQ0017
C        D CONTAINS THE DIAGONAL ELEMENTS OF THE INPUT MATRIX;       IMTQ0018
C                                                                    IMTQ0019
C        E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE INPUT MATRIX     IMTQ0020
C          IN ITS FIRST N-1 POSITIONS.  E(N) IS ARBITRARY;           IMTQ0021
C                                                                    IMTQ0022
C        Z CONTAINS THE FIRST ROW OF THE IDENTITY MATRIX.            IMTQ0023
C                                                                    IMTQ0024
C     ON OUTPUT:                                                     IMTQ0025
C                                                                    IMTQ0026
C        D CONTAINS THE EIGENVALUES IN ASCENDING ORDER.  IF AN       IMTQ0027
C          ERROR EXIT IS MADE, THE EIGENVALUES ARE CORRECT BUT       IMTQ0028
C          UNORDERED FOR INDICES 1, 2, ..., IERR-1;                  IMTQ0029
C                                                                    IMTQ0030
C        E HAS BEEN DESTROYED;                                       IMTQ0031
C                                                                    IMTQ0032
C        Z CONTAINS THE FIRST COMPONENTS OF THE ORTHONORMAL EIGENVECTORSIMTQ0033
C          OF THE SYMMETRIC TRIDIAGONAL MATRIX.  IF AN ERROR EXIT IS IMTQ0034
C          MADE, Z CONTAINS THE EIGENVECTORS ASSOCIATED WITH THE STORED IMTQ0035
C          EIGENVALUES;                                              IMTQ0036
C                                                                    IMTQ0037
C        IERR IS SET TO                                              IMTQ0038
C          ZERO       FOR NORMAL RETURN,                            IMTQ0039
C          J          IF THE J-TH EIGENVALUE HAS NOT BEEN           IMTQ0040
C                     DETERMINED AFTER 30 ITERATIONS.               IMTQ0041
C                                                                    IMTQ0042
```

Appendix A (contd.)

```
C                                                                       IMTQ0043
      INTEGER I, J, K, L, M, N, II, MML, IERR                           IMTQ0044
      REAL*8 D(N), E(N), Z(N), B, C, F, G, P, S, MACHEP                 IMTQ0045
      REAL*8 DSQRT, DABS, DSIGN                                         IMTQ0046
C                                                                       IMTQ0047
C     :::::::::: MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING     IMTQ0048
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.   IMTQ0049
C                MACHEP = 16D0 ** (-13) FOR LONG FORM ARITHMETIC        IMTQ0050
C                ON S360 ::::::::::                                     IMTQ0051
      DATA MACHEP/Z3410000000000000/                                    IMTQ0052
C                                                                       IMTQ0053
      IERR = 0                                                          IMTQ0054
      IF (N .EQ. 1) GO TO 1001                                          IMTQ0055
C                                                                       IMTQ0056
      E(N) = 0D0                                                        IMTQ0057
      DO 240 L = 1, N                                                   IMTQ0058
         J = 0                                                          IMTQ0059
C     :::::::::: LOOK FOR SMALL SUB-DIAGONAL ELEMENT ::::::::::        IMTQ0060
  105    DO 110 M = L, N                                                IMTQ0061
            IF (M .EQ. N) GO TO 120                                     IMTQ0062
            IF (DABS(E(M)) .LE. MACHEP * (DABS(D(M)) + DABS(D(M+1))))  IMTQ0063
     X         GO TO 120                                                IMTQ0064
  110    CONTINUE                                                       IMTQ0065
C                                                                       IMTQ0066
  120    P = D(L)                                                       IMTQ0067
         IF (M .EQ. L) GO TO 240                                        IMTQ0068
         IF (J .EQ. 30) GO TO 1000                                      IMTQ0069
         J = J + 1                                                      IMTQ0070
C     :::::::::: FORM SHIFT ::::::::::                                  IMTQ0071
         G = (D(L+1) - P) / (2D0 * E(L))                                IMTQ0072
         R = DSQRT(G*G+1D0)                                             IMTQ0073
         G = D(M) - P + E(L) / (G + DSIGN(R, G))                        IMTQ0074
         S = 1D0                                                        IMTQ0075
         C = 1D0                                                        IMTQ0076
         P = 0D0                                                        IMTQ0077
         MML = M - L                                                    IMTQ0078
C                                                                       IMTQ0079
C     :::::::::: FOR I=M-1 STEP -1 UNTIL L DO -- ::::::::::            IMTQ0080
         DO 200 II = 1, MML                                             IMTQ0081
            I = M - II                                                  IMTQ0082
            F = S * E(I)                                                IMTQ0083
            B = C * E(I)                                                IMTQ0084
            IF (DABS(F) .LT. DABS(G)) GO TO 150                         IMTQ0085
            C = G / F                                                   IMTQ0086
            R = DSQRT(C*C+1D0)                                          IMTQ0087
            E(I+1) = F * R                                              IMTQ0088
            S = 1D0 / R                                                 IMTQ0089
            C = C * S                                                   IMTQ0090
            GO TO 160                                                   IMTQ0091
  150       S = F / G                                                   IMTQ0092
            R = DSQRT(S*S+1D0)                                          IMTQ0093
            E(I+1) = G * R                                              IMTQ0094
            C = 1D0 / R                                                 IMTQ0095
            S = S * C                                                   IMTQ0096
```

Appendix A (contd.)

```
  160       G = D(I+1) - P                                                IMTQ0097
            R = (D(I) - G) * S + 2D0 * C * B                              IMTQ0098
            P = S * R                                                     IMTQ0099
            D(I+1) = G + P                                                IMTQ0100
            G = C * R - B                                                 IMTQ0101
C     :::::::::: FORM FIRST COMPONENT OF VECTOR ::::::::::                IMTQ0102
            F = Z(I+1)                                                    IMTQ0103
            Z(I+1) = S * Z(I) + C * F                                     IMTQ0104
  200       Z(I) = C * Z(I) - S * F                                       IMTQ0105
C                                                                         IMTQ0106
         D(L) = D(L) - P                                                  IMTQ0107
         E(L) = G                                                         IMTQ0108
         E(M) = 0D0                                                       IMTQ0109
         GO TO 105                                                        IMTQ0110
  240 CONTINUE                                                            IMTQ0111
C                                                                         IMTQ0112
C     :::::::::: ORDER EIGENVALUES AND EIGENVECTORS ::::::::::            IMTQ0113
      DO 300 II = 2, N                                                    IMTQ0114
         I = II - 1                                                       IMTQ0115
         K = I                                                            IMTQ0116
         P = D(I)                                                         IMTQ0117
C                                                                         IMTQ0118
         DO 260 J = II, N                                                 IMTQ0119
            IF (D(J) .GE. P) GO TO 260                                    IMTQ0120
            K = J                                                         IMTQ0121
            P = D(J)                                                      IMTQ0122
  260    CONTINUE                                                         IMTQ0123
C                                                                         IMTQ0124
         IF (K .EQ. I) GO TO 300                                          IMTQ0125
         D(K) = D(I)                                                      IMTQ0126
         D(I) = P                                                         IMTQ0127
         P = Z(I)                                                         IMTQ0128
         Z(I) = Z(K)                                                      IMTQ0129
         Z(K) = P                                                         IMTQ0130
  300 CONTINUE                                                            IMTQ0131
C                                                                         IMTQ0132
      GO TO 1001                                                          IMTQ0133
C     :::::::::: SET ERROR -- NO CONVERGENCE TO AN                        IMTQ0134
C                EIGENVALUE AFTER 30 ITERATIONS ::::::::::              IMTQ0135
 1000 IERR = L                                                            IMTQ0136
 1001 RETURN                                                              IMTQ0137
C     :::::::::: LAST CARD OF IMTQL2 ::::::::::                           IMTQ0138
      END                                                                 IMTQ0139
C                                                                         QNC80001
      SUBROUTINE QUANC8( FUN, A, B, ABSERR, RELERR, RESULT, ERREST,       QNC80002
     1                                                  NOFUN, FLAG)      QNC80003
C   ESTIMATE THE INTEGRAL OF FUN(X) FROM A TO B TO A USER PROVIDED        QNC80004
C   TOLERANCE:  AN AUTOMATIC ADAPTIVE ROUTINE BASED ON THE 8-PANEL        QNC80005
C   NEWTON-COTES RULE.                                                    QNC80006
C                                                                         QNC80007
```

Appendix A (contd.)

```
C   INPUT :                                                              QNC80008
C                                                                        QNC80009
C   FUN     THE NAME OF THE INTEGRAND FUNCTION SUBPROGRAM FUN(X).        QNC80010
C           FUNCTION NAME SHOULD APPEAR IN AN EXTERNAL STATEMENT IN      QNC80011
C           CALLING PROGRAM                                              QNC80012
C                                                                        QNC80013
C   A       THE LOWER LIMIT OF INTEGRATION.                              QNC80014
C                                                                        QNC80015
C   B       THE UPPER LIMIT OF INTEGRATION.(B MAY BE LESS THAN A.)       QNC80016
C                                                                        QNC80017
C   RELERR  A RELATIVE ERROR TOLERANCE. (SHOULD BE NON-NEGATIVE)         QNC80018
C                                                                        QNC80019
C   ABSERR  AN ABSOLUTE ERROR TOLERANCE. (SHOULD BE NON-NEGATIVE)        QNC80020
C                                                                        QNC80021
C   OUTPUT:                                                              QNC80022
C                                                                        QNC80023
C   RESULT  AN APPROXIMATION TO THE INTEGRAL HOPEFULLY SATISFYING THE    QNC80024
C           LEAST STRINGENT OF THE TWO ERROR TOLERANCES.                 QNC80025
C                                                                        QNC80026
C   ERREST  AN ESTIMATE OF THE MAGNITUDE OF THE ACTUAL ERROR.            QNC80027
C                                                                        QNC80028
C   NOFUN   THE NUMBER OF FUNCTION VALUES USED IN CALCULATION OF RESULT.QNC80029
C                                                                        QNC80030
C   FLAG    A RELIABILITY INDICATOR.  IF FLAG IS ZERO, THEN RESULT       QNC80031
C           PROBABLY SATISFIES THE ERROR TOLERANCE.  IF FLAG IS          QNC80032
C           XXX.YYY , THEN  XXX = THE NUMBER OF INTERVALS WHICH HAVE     QNC80033
C           NOT CONVERGED AND  0.YYY = THE FRACTION OF THE INTERVAL      QNC80034
C           LEFT TO DO WHEN THE LIMIT ON  NOFUN  WAS APPROACHED.         QNC80035
C                                                                        QNC80036
C   REFERENCE:                                                           QNC80037
C                                                                        QNC80038
C      G. E. FORSYTHE, M. A. MALCOLM, AND C. B. MOLER, COMPUTER          QNC80039
C      METHODS FOR MATHEMATICAL COMPUTATIONS (PRENTICE-HALL, ENGLEWOOD   QNC80040
C      CLIFFS, N. J., 1977), PP. 84-105.                                 QNC80041
C                                                                        QNC80042
      DOUBLE PRECISION FUN, A, B, ABSERR, RELERR, RESULT, ERREST, FLAG   QNC80043
      INTEGER NOFUN                                                      QNC80044
      DOUBLE PRECISION W0,W1,W2,W3,W4,AREA,X0,F0,STONE,STEP,COR11,TEMP   QNC80045
      DOUBLE PRECISION QPREV,QNOW,QDIFF,QLEFT,ESTERR,TOLERR              QNC80046
      DOUBLE PRECISION QRIGHT(31),F(16),X(16),FSAVE(8,30),XSAVE(8,30)    QNC80047
      DOUBLE PRECISION DABS,DMAX1                                        QNC80048
      INTEGER LEVMIN,LEVMAX,LEVOUT,NOMAX,NOFIN,LEV,NIM,I,J               QNC80049
C                                                                        QNC80050
C   *** STAGE 1 ***  GENERAL INITIALIZATION                              QNC80051
C   SET CONSTANTS.                                                       QNC80052
C                                                                        QNC80053
      LEVMIN = 1                                                         QNC80054
      LEVMAX = 30                                                        QNC80055
      LEVOUT = 6                                                         QNC80056
      NOMAX = 5000                                                       QNC80057
      NOFIN = NOMAX - 8*(LEVMAX-LEVOUT+2**(LEVOUT+1))                    QNC80058
C                                                                        QNC80059
```

Appendix A (contd.)

```
C   TROUBLE WHEN NOFUN REACHES NOFIN                                    QNC80060
C                                                                       QNC80061
      W0 =   3956D0 / 14175D0                                           QNC80062
      W1 =  23552D0 / 14175D0                                           QNC80063
      W2 =  -3712D0 / 14175D0                                           QNC80064
      W3 =  41984D0 / 14175D0                                           QNC80065
      W4 = -13160D0 / 14175D0                                           QNC80066
C                                                                       QNC80067
C   INITIALIZE RUNNING SUMS TO ZERO.                                    QNC80068
C                                                                       QNC80069
      FLAG = 0D0                                                        QNC80070
      RESULT = 0D0                                                      QNC80071
      COR11  = 0D0                                                      QNC80072
      ERREST = 0D0                                                      QNC80073
      AREA   = 0D0                                                      QNC80074
      NOFUN = 0                                                         QNC80075
      IF (A .EQ. B) RETURN                                              QNC80076
C                                                                       QNC80077
C   ***   STAGE 2 ***   INITIALIZATION FOR FIRST INTERVAL              QNC80078
C                                                                       QNC80079
      LEV = 0                                                           QNC80080
      NIM = 1                                                           QNC80081
      XC = A                                                            QNC80082
      X(16) = B                                                         QNC80083
      QPREV  = 0D0                                                      QNC80084
      F0 = FUN(X0)                                                      QNC80085
      STONE = 0.0625D0 * (B - A)                                        QNC80086
      X(8)  =  0.5D0 * (X0  + X(16))                                    QNC80087
      X(4)  =  0.5D0 * (XC  + X(8))                                     QNC80088
      X(12) =  0.5D0 * (X(8)  + X(16))                                  QNC80089
      X(2)  =  0.5D0 * (X0  + X(4))                                     QNC80090
      X(6)  =  0.5D0 * (X(4)  + X(8))                                   QNC80091
      X(10) =  0.5D0 * (X(8)  + X(12))                                  QNC80092
      X(14) =  0.5D0 * (X(12) + X(16))                                  QNC80093
      DO 25 J = 2, 16, 2                                                QNC80094
         F(J) = FUN(X(J))                                               QNC80095
   25 CONTINUE                                                          QNC80096
      NOFUN = 9                                                         QNC80097
C                                                                       QNC80098
C   ***   STAGE 3 ***   CENTRAL CALCULATION                            QNC80099
C   REQUIRES QPREV,X0,X2,X4,...,X16,F0,F2,F4,...,F16.                   QNC80100
C   CALCULATES X1,X3,...X15, F1,F3,...F15,QLEFT,QRIGHT,QNOW,QDIFF,AREA. QNC80101
   30 X(1) = 0.5D0 * (XC + X(2))                                        QNC80102
      F(1) = FUN(X(1))                                                  QNC80103
      DO 35 J = 3, 15, 2                                                QNC80104
               X(J) = 0.5D0 * (X(J-1) + X(J+1))                         QNC80105
         F(J) = FUN(X(J))                                               QNC80106
   35 CONTINUE                                                          QNC80107
      NOFUN = NOFUN + 8                                                 QNC80108
      STEP = 0.0625D0 * (X(16) - XC)                                    QNC80109
      QLEFT  =  (W0*(F0 + F(8))  + W1*(F(1)+F(7))  + W2*(F(2)+F(6))     QNC80110
     1                           + W3*(F(3)+F(5))  +  W4*F(4)) * STEP   QNC80111
```

Appendix A (contd.)

```
      QRIGHT(LEV+1) = (W0*(F(8)+F(16))+W1*(F(9)+F(15))+W2*(F(10)+F(14)) QNC80112
     1                          + W3*(F(11)+F(13)) + W4*F(12)) * STEP   QNC80113
      QNOW = QLEFT + QRIGHT(LEV+1)                                      QNC80114
      QDIFF = QNOW - QPREV                                              QNC80115
      AREA = AREA + QDIFF                                               QNC80116
C                                                                       QNC80117
C   ***   STAGE 4 *** INTERVAL CONVERGENCE TEST                         QNC80118
      ESTERR = DABS(QDIFF) / 1023D0                                     QNC80119
      TOLERR = DMAX1(ABSERR,RELERR*DABS(AREA)) * (STEP/STONE)           QNC80120
      IF (LEV .LT. LEVMIN) GO TO 50                                     QNC80121
      IF (LEV .GE. LEVMAX) GO TO 62                                     QNC80122
      IF (NOFUN .GT. NOFIN) GO TO 60                                    QNC80123
      IF (ESTERR .LE. TOLERR) GO TO 70                                  QNC80124
C                                                                       QNC80125
C   ***   STAGE 5   ***   NO CONVERGENCE                               QNC80126
C   LOCATE NEXT INTERVAL.                                               QNC80127
   50 NIM = NIM + NIM                                                   QNC80128
      LEV = LEV + 1                                                     QNC80129
C                                                                       QNC80130
C   STORE RIGHT HAND ELEMENTS FOR FUTURE USE.                           QNC80131
      DO 52 I = 1, 8                                                    QNC80132
         FSAVE(I,LEV) = F(I+8)                                          QNC80133
         XSAVE(I,LEV) = X(I+8)                                          QNC80134
   52 CONTINUE                                                          QNC80135
C                                                                       QNC80136
C   ASSEMBLE LEFT HAND ELEMENTS FOR IMMEDIATE USE.                      QNC80137
      QPREV = QLEFT                                                     QNC80138
      DO 55 I = 1, 8                                                    QNC80139
         J = - I                                                        QNC80140
         F(J+J+18) = F(J+9)                                             QNC80141
         X(J+J+18) = X(J+9)                                             QNC80142
   55 CONTINUE                                                          QNC80143
      GO TO 30                                                          QNC80144
C                                                                       QNC80145
C   ***   STAGE 6   ***   TROUBLE SECTION                              QNC80146
C   NUMBER OF FUNCTION VALUES IS ABOUT TO EXCEED LIMIT.                 QNC80147
   60 NOFIN = NOFIN + NOFIN                                             QNC80148
      LEVMAX = LEVOUT                                                   QNC80149
      FLAG = FLAG + (B - XC) / (B - A)                                  QNC80150
      GO TO 70                                                          QNC80151
C                                                                       QNC80152
C   CURRENT LEVEL IS LEVMAX.                                            QNC80153
   62 FLAG = FLAG + 1D0                                                 QNC80154
C                                                                       QNC80155
C   ***   STAGE 7   ***   INTERVAL CONVERGED                           QNC80156
C   ADD CONTRIBUTIONS INTO RUNNING SUMS.                                QNC80157
   70 RESULT = RESULT + QNOW                                            QNC80158
      ERREST = ERREST + ESTERR                                          QNC80159
      COR11  = COR11  + QDIFF / 1023D0                                  QNC80160
C                                                                       QNC80161
C   LOCATE NEXT INTERVAL.                                               QNC80162
C                                                                       QNC80163
```

Appendix A (contd.)

```
   72 IF (NIM .EQ. 2*(NIM/2)) GO TO 75                                  QNC80164
      NIM = NIM / 2                                                     QNC80165
      LEV = LEV - 1                                                     QNC80166
      GO TO 72                                                          QNC80167
   75 NIM = NIM + 1                                                     QNC80168
      IF (LEV .LE. 0) GO TO 80                                          QNC80169
C                                                                       QNC80170
C   ASSEMBLE ELEMENTS REQUIRED FOR THE NEXT INTERVAL.                   QNC80171
      QPREV = QRIGHT(LEV)                                               QNC80172
      X0 = X(16)                                                        QNC80173
      F0 = F(16)                                                        QNC80174
      DO 78 I = 1, 8                                                    QNC80175
         F(I+I) = FSAVE(I,LEV)                                          QNC80176
         X(I+I) = XSAVE(I,LEV)                                          QNC80177
   78 CONTINUE                                                          QNC80178
      GO TO 30                                                          QNC80179
C                                                                       QNC80180
C   ***   STAGE 8   ***   FINALIZE AND RETURN                          QNC80181
   80 RESULT = RESULT + COR11                                           QNC80182
C                                                                       QNC80183
C   MAKE SURE ERREST NOT LESS THAN ROUNDOFF LEVEL.                      QNC80184
      IF (ERREST .EQ. CD0) RETURN                                       QNC80185
   82 TEMP = DABS(RESULT) + ERREST                                      QNC80186
      IF (TEMP .NE. DABS(RESULT)) RETURN                                QNC80187
      ERREST = ERREST + ERREST                                          QNC80188
      GO TO 82                                                          QNC80189
      END                                                               QNC80190
```

Appendix B

ANISN Sputtering Calculations Using MUXS Data

Three changes were made to the standard version of ANISN in order to perform sputtering calculations with the cross sections generated by MUXS:

1. From Eq. 1, $\Sigma_{G-1}^{CSD}\,\phi_{G-1}(x,\mu)$ must be added to the total source term for group G if G > 1. We have made this change to ANISN in Subroutine S833. Two arrays are used: E3SAV1(I,M) and E3SAV2(I,M) where I is an index for the spatial mesh interval and M is an index for the angular quadrature interval. E3SAV1 saves $\Sigma_{G}^{CSD}\,\phi_{G}(x,\mu)$ for the current energy group being processed by S833. ESAVE2 is the value of $\Sigma_{G-1}^{CSD}\,\phi_{G-1}(x,\mu)$ obtained from the previous call to S833. ESAV2 is added to the total source term, XNRI, if the group being processed is not group 1.

2. An angle dependent response function for the leakage is needed to prevent leakage by target atoms with energy less than U_o/μ^2, where U_o is the surface binding energy and μ is the cosine of the angle between the surface normal and the target atom direction. This is essentially a post-editing procedure and would require no modification to ANISN if a spherically symmetric surface binding energy were assumed.[13] We have made this change to Subroutine SUMARY of ANISN. The current at the right surface of the slab, XND, is multiplied by this response function, R, to obtain the right leakage for the summary

tables, i.e., RCT. The ion source is assumed to have been input on the right surface of the slab by the user. With this change, the calculated sputtering yield will be printed as a function of the energy group in the summary tables under the heading "RT LEAKAGE."

3. We have also modified Subroutine S833 to input Σ_G^{CSD}, SIGSD, and the target atom energy group structure, EBT, from the cross section file produced by MUXS. With this change, all card input to ANISN is the same as that for the standard version.

The modified ANISN routines, S833 and SUMARY, along with the required data and JCL for a problem with an 8 kev proton incident on a nickel target, are shown in Figure B1. The statements that were added are indented and commented.

This problem was run with the cross sections obtained from MUXS using the input described in Section V. The calculated sputtering yield is 0.00818 Ni atom/incident proton. This result is in excellent agreement with the experimental value (.00822 Ni atom/incident proton) obtained by Roth, Bohdansky, and Ottenberger.[14]

```
//TJHANSP JCB (21541,I6),'6025 T J HOFFMAN',TIME=6
//STP3 EXEC FORTHCLG,PARM.FORT='XREF',PARM.LKED='MAP,LET,LIST',
//  GOSIZE=950K
C  MAIN ROUTINE - ALLOT SPACE ( LIM1 AND DUMY )
      COMMON/BULKBU/D(1),LIM1,DUMY(170000)
      LIM1=170000
      CALL CLEAR(3,LIM1)
      CALL CONTRL
      STOP
      END
      SUBROUTINE S833(XN,XNN,ST,SR,CS,MA,J5,AA,SAT,SA,CH,XNA,XNR,PNC,DSNANSN2192
     1 ,W,XNI, NO,B,DA,DB,DC,DS,V,XNE,PA,WD,CT,SG,XND,I1,   M1,I2,IM1,MRANSN2193
     1,ART,ALFT)                                                      ANSN2194
C   *** ITER  *IMPOSSIBLE TO *TRANSLATE, *EXPLAIN OR *REMEMBER       ANSN219
C SPUTTERING CHANGE - SETUP ARRAYS, COMMON/SPUT/
        REAL*8 EBT,UO
        DIMENSION E3SAV1(201,33),E3SAV2(201,33)
        COMMON/SPUT/SIGSD(50),EET(51),UO,NI
      DIMENSION XN(1)    ,XNN(1),ST(1),SR(1),CS(1),MA(1),J5(1),AA(1),    ANSN2196
     1  SAT(1)    ,SA(1)    ,CH(1)    ,XNA(1)    ,XNR(1),PNC(1)    ,DSN(1),  ANSN21
     2 W(1),XNI(1), NO(1),B(1)    ,DA(1)    ,DB(1)    ,DC(1)    ,DS(1)    ,  ANSN2198
     3 V(1),XNE(1),PA(1)        ,WD(1),CT(1),SG(1),XND(1)    ,MR(1)        ANSN2199
     4 ,ART(1),ALFT(1)                                                ANSN2200
      COMMON /BULKBU/                                                 ANSN2201
     $       D(1),LIM1,LXKI,LFD,LXN,LR,LVE,LW,LDSN,LMA,LMZ,LMB,LMC,LXMD,ANSN2202
     1LMTT,LCRX,LFIX,LFLT,LQ,LPA,LJ5,LRM,LDF,LJ3,LJ4,LIGT,LART,LALFT,   ANSN2203
     2LFGP,LFGG,LEND,LV,LAA,LWD,LMR,LPNC,LXJ,LCH,LCA,LCF,LCT,LCS,LTAB,  ANSN2204
     3LXND,LSA,LSAT,LRAV,LRA,LXNN,LXNF,LXNR,LXNA,LSR,LST,LQG,LFG,LSG,   ANSN2205
     4LXKE,LXNI,LXNC,LT3,LT5,LDA,LDB,LDC,LDS,LB,IGMP,IGMM,IIGG,NERR,IMJTANSN2206
     5,IHG,IMP,MP,NDS,NUS,SDG,SCG,AG,XNLGG,XNLG,SNG,ALA,ASR,EAM,EPG,EQ,  ANSN2207
     6E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,E11,E12,E13,E14,E15,E16,E17,E18,E19ANSN2208
     7,E20,ESC,ESM,EVP,EVPP,ETP,IC,ICVT,IGP,IG,IHP,IIC,IIG,IP,IZP,IO1,   ANSN2209
     8IO2,IO3,IO4,IO5,IO6,IO7,IO8,IO9,IOG,JT,LC,MG,MI,ML,MM,NFN,XITR,    ANSN2210
     9XLAP,XLAPP,XLAR,XLA,XNIO,XNII,ZZ1,ZZ2,ZZ3,XNB,XKEP,XKIP,IH,I,K,L,  ANSN2211
     AM,J,N,NN,ISV,                                                   ANSN2212
     BID,ITH,ISCT,ISN,IGP,IBL,IBR,IZM,IM,IPVT,IGM,IPT,IHS,IHM,MS,MCR,MTPANSN2213
     C,MT,IDFM,IPVT,IQM,IPM,IPP,IIM,ID1,ID2,ID3,ID4,ICM,IDAT1,IDAT2,IFG,ANSN2214
     DIFLU,IFN,IPRT,IXTR,                                             ANSN2215
     EEV,EVM,EPS,BF,DY,DZ,DFM1,XNF,PV,PYF,XLAL,YLAM,EQL,XNPM,         ANSN2216
     ET(12),NIN,NOU,NT1,NT2,NT3,NT4,NT5,NT6,NT7                       ANSN2217
C SPUTTERING CHANGE - INPUT DATA
        DATA ICALL/0/
        IF(ICALL.NE.0) GO TO 200
 210    READ(4) N1,N2,IDENT
        IF(IDENT.EQ.7) GO TO 220
        READ(4) DUM
        GO TO 210
 220    READ(4) (SIGSD(IGG),IGG=1,IGM)
        DO 230 IGG=1,IGM
        IF(SIGSD(IGG).EQ.0.) GO TO 240
 230    CONTINUE
        NI=IGM
        GO TO 250
 240    NI=IGG
 250    NTP1=IGM-NI+1
```

Figure B1. ANISN Sample Problem Input

```
      GO TO 250
240   NI=IGG
250   NTP1=ICM-NI+1
      READ(4) (EBT(IGG),IGG=1,NTP1)
      UC=EBT(NTP1)
      ICALL=1
260   CONTINUE
      IM2=IM+IM                                                    ANSN2218
      E1=0.0                                                       ANSN2219
      ICMC=1                                                       ANSN2220
      DO 8003 I=1,IM                                               ANSN2221
      E1=E1 + CT(I)*DC(I)                                          ANSN2222
      NO(I)=ICMC                                                   ANSN2223
      IF(I.LT.IM .AND. E1.LT.1.0)GO TO 8003                        ANSN2224
      NO(ICMC + IM)=I+1                                            ANSN2225
      E1=0.0                                                       ANSN2226
      ICMC=ICMC + 1                                                ANSN2227
8003  CONTINUE                                                     ANSN2228
      ICMC=NO(IM)                                                  ANSN2229
      NGE=0                                                        ANSN2230
      IF(IGE.EQ.2)NGE=1                                            ANSN2231
      IFLU1=IFLU                                                   ANSN2232
      IF(IFLU.EQ.4)IFLU1=0                                         ANSN2233
      IGMPM=(IGMP-1)*MM                                            ANSN2234
      IGMPI=(IGMP-1)*IM                                            ANSN2235
      IGMI=(IIGG-1)*MM*IMP - MM                                    ANSN2236
      IF(IC.GT.1  .OR.  IC.EQ.1 .AND. IFN.GE.1) GO TO 832          ANSN2237
C     *** FIRST GENERATION BOUNDARY FLUXES                         ANSN223
      IGMPMM=IGMPM + 1                                             ANSN2239
      DO 55 M=1,MM                                                 ANSN2240
      IF(DSN(M))53,53,54                                           ANSN2241
   53 B(IGMPMM)=XNN(1)                                             ANSN2242
      GO TO 55                                                     ANSN2243
   54 B(IGMPMM)=XNN(IM)                                            ANSN2244
   55 IGMPMM=IGMPMM + 1                                            ANSN2245
  832 IGMPII=IGMPI + 1                                             ANSN2246
      SCLMI=0.0                                                    ANSN2247
      IF(IIC.GT.IIM/2 .AND. IIM.GE.20)IFLU1=3                      ANSN224
      DO 1001 I=1,IM                                               ANSN2249
      XN(IGMPII)=XNN( )                                            ANSN2250
      IGMPII=IGMPII + 1                                            ANSN2251
C     *** ADD IN-GROUP SOURCE COMPONENT (PO)                       ANSN2252
       ST(I)=SR(I)+ CS(I)*XNN(I)                                   ANSN2253
1001  XNN(I)=0.0                                                   ANSN2254
      IF(ISCT.EQ.0)GO TO 1002                                      ANSN2255
      DO 1    I=1,IM                                               ANSN2256
      NLO=1                                                        ANSN2257
      J=MA(I)                                                      ANSN2258
      J=J5(J)                                                      ANSN2259
      IF(J.EQ.0)GO TO 1                                            ANSN2260
      INN=I                                                        ANSN2261
      IN=I                                                         ANSN22 2
```

Figure B1 (contd.)

```
      DO 2 N=1,J                                                    ANSN2263
      NHI=NLO + NGE*(N+N+1)/4                                       ANSN2264
      DO 3 NN=NLO,NHI                                               ANSN2265
      SAT(INN)=SA(INN) + CH(IN)*XNA(INN)                            ANSN2266
      XNA(INN)=0.0                                                  ANSN2267
    3 INN=INN+IM                                                    ANSN2268
      NLO=NHI+1                                                     ANSN2269
    2 IN=IN+IM                                                      ANSN2270
    1 CONTINUE                                                      ANSN2271
 1002 CONTINUE                                                      ANSN2272
      LC=LC+1                                                       ANSN2273
      KK=MM+1                                                       ANSN2274
      IF(IEL.NE.3 .AND. IBR.NE.3)GO TO 842                          ANSN2275
C   *** COMPUTE WHITE BOUNDARY FLUX                                 ANSN2276
      XNII=0.0                                                      ANSN2277
      XNIO=0.0                                                      ANSN2278
      IGMPMM=IGMPM + 1                                              ANSN2279
      DO 45 M=1,MM                                                  ANSN2280
      IF(WD(M))46,46,47                                             ANSN2281
   46 XNII=XNII + B(IGMPMM)*ABS(WD(M))                              ANSN2282
      GO TO 45                                                      ANSN2283
   47 XNIO=XNIO + B(IGMPMM)*WD(M)                                   ANSN2284
   45 IGMPMM=IGMPMM + 1                                             ANSN2285
      IF(IBL.EQ.3)XNII=XNII*ALFT(IIG)/ZZ3                           ANSN2286
      IF(IBR.EQ.3)XNIO=XNIO*ART(IIG)/ZZ3                            ANSN2287
C   *** BEGIN ANGLE LOOP FOR FLUX CALCULATION                       ANSN22
  842 DO 5 M=1,MM                                                   ANSN2289
      DSNM=DSN(M)                                                   ANSN2290
      WM=W(M)                                                       ANSN2291
      IPMM=(M-1)*IP                                                 ANSN2292
      IGMIM=IGMI + M                                                ANSN2293
C   *** COMBINE PC AND PL SOURCE TERMS                              ANSN2294
      DO 6 I= 1, IM                                                 ANSN2295
    6 XNR(I) = ST(I)                                                ANSN2296
      IF(ISCT.EQ.0)GO TO 9                                          ANSN2297
      MN=M                                                          ANSN2298
      IN=1                                                          ANSN2299
      DO 8 N=1,JT                                                   ANSN2300
      DO 88 I=1,IM                                                  ANSN2301
      XNR(I)=XNR(I) + PNC(MN)*SAT(IN)                               ANSN2302
   88 IN=IN + 1                                                     ANSN2303
    8 MN=MN + MM                                                    ANSN2304
C   *** COMPUTE TRANSVERSE VOID STREAMING CORRECTION                ANSN2305
    9 E0=0.0                                                        ANSN2306
      IF(DFM1.EQ.0.0)GO TO 10                                       ANSN2307
      IF(IGE.EQ.1)E0=SQRT(1.0 - DSNM*DSNM)/(0.5*DFM1)               ANSN2308
      IF(IGE.NE.2)GO TO 10                                          ANSN2309
      IF(WM.NE.0)GO TO 11                                           ANSN2310
      KK=KK-1                                                       ANSN2311
   11 E0=DSN(KK)/(0.5*DFM1)                                         ANSN2312
C   *** SET UP BOUNDARY CONDITIONS                                  ANSN2313
   10 IF(DSNM.GE.0.0)GO TO 12                                       ANSN2314
```

Figure B1 (contd.)

```
      ILS=IP                                                          ANSN2315
      IT=IF                                                           ANSN2316
      I00=IP + 1                                                      ANSN2317
      I01=IPR                                                         ANSN2318
      E1=XNIC                                                         ANSN2319
      GO TO 19                                                        ANSN2320
   12 IT=0                                                            ANSN2321
      ILS=-1                                                          ANSN2322
      I00=0                                                           ANSN2323
      I01=IPL                                                         ANSN2324
      E1=XNII                                                         ANSN2325
   19 IF(I01-1)13,14,15                                               ANSN2326
   13 XNM=0.0                                                         ANSN2327
      GO TO 17                                                        ANSN2328
   14 IGMPML=IGMPM + MR(M)                                            ANSN2329
      XNM=B(IGMPML)                                                   ANSN2330
      GO TO 17                                                        ANSN2331
   15 XNM=E1                                                          ANSN2332
      IGMPMM=IGMPM + M                                                ANSN2333
      IF(I01.EQ.2)XNM=B(IGMPMM)                                       ANSN2334
   17 IF(WM.NE.0.0)GO TO 24                                           ANSN2335
      DO 18 I=1,IM                                                    ANSN2336
   18 XNE(I)=0.0                                                      ANSN2337
   24 E6=XNM                                                          ANSN2338
C   *** BEGIN INTERVAL LOOP FOR FLUX CALCULATION                      ANSN2339
      IF(IPM.NE.1)GO TO 8844                                          ANSN2340
      PAIGMM=PA(IGMIM + MM)                                           ANSN2341
C .....NOTE.....
C.....SOME PATCHWORK CODING NOW IN S833
C.............ALSO NOTE FIXED DIMENSION STMT. IN THIS ROUTINE <<<REDO LATER>>
 8844 DO 25 J=1,IM                                                    ANSN2342
  844 L =IABS(I00-J)                                                  ANSN2343
  843 I =IABS(IT-J)                                                   ANSN2344
      K=IABS(ILS-J)                                                   ANSN2345
      IPMMI=IPMM+I                                                    ANSN2346
      IPMMK=IPMM+K                                                    ANSN2347
      IPMML=IPMM + L                                                  ANSN2348
      IF(DSNM.GT.0.0)GO TO 26                                         ANSN2349
C   *** ADD SHELL SOURCE TO FLUX WHEN MU.LT.0                         ANSN2350
      IF(IPM-1)26,27,28                                               ANSN2351
   28 IGMIMI=IGMIM + I*MM                                             ANSN2352
      XNM=XNM + PA(IGMIMI)                                            ANSN2353
      GO TO 26                                                        ANSN2354
   27 IF(IPP.EQ.I)XNM=XNM + PAIGMM                                    ANSN2355
C   *** SAVE ANGULAR FLUX                                             ANSN2356
   26 XND(IPMML)=E6                                                   ANSN2357
C  * * * * * * * * * *   ANISN FIVE MODE FLUX CALCULATION            *ANSN2358
C  * * * * * * * * * *   E3 = CENTERED FLUX                         *ANSN2359
C  * * * * * * * * * *   E5 = ANGLE EXTRAPOLATED FLUX               *ANSN2360
C  * * * * * * * * * *   E6 = SPACE EXTRAPOLATED FLUX               *ANSN2361
C  * * * * * * * * * *   XNM = KNOWN SPACE BOUNDARY FLUX            *ANSN2362
C  * * * * * * * * * *   PHIM = KNOWN ANGLE BOUNDARY FLUX           *ANSN2363
```

Figure B1 (contd.)

```
C  * * * * * * * * * *   XNRI = TOTAL SOURCE                          *ANSN2364
C  * * * * * * * * * *   E1 = SIGMA TOTAL                            *ANSN2365
   56 E1=CT(I)                                                        ANSN2366
      IF(E1.EQ.0.0)E1=E1 + E6*V(I)                                    ANSN2367
      XNRI=XNR(I)                                                     ANSN236
C SPUTTERING CHANGE - ADD CSD TERM TO TOTAL SOURCE
        IF(IIG.NE.1) XNRI=XNRI+E3SAV2(I,M)*V(I)
      DSKM=DS(IPMMK)                                                  ANSN2369
      DSLM=DS(IPMML)                                                  ANSN2370
      DSKM2=DSKM + DSLM                                               ANSN2371
      IF(WM.EQ.C.0 .OR. IGE.EQ.1)GO TO 6006                           ANSN2372
C  * * * * * * * * * *   GENERAL DIRECTION FLUX CALCULATION  -  W.GT.0 *ANSN2373
      PHIM=XNE(I)                                                     ANSN2374
      DAIM=DA(IPMMI)                                                  ANSN2375
      DBIM=DB(IPMMI)                                                  ANSN2376
      DBIM2=DAIM + DBIM                                               ANSN2377
      IF(IFLU1-2)6001,6002,6003                                       ANSN237
C  * * * * * * * * * *   LINEAR MODEL                                 *ANSN2379
 6001 E3=(XNRI + DSKM2*XNM + DBIM2*PHIM)/(E1 + DSKM2 + DBIM2)         ANSN23 0
      E5=E3 + E3 - PHIM                                               ANSN2381
      E6=E3 + E3 - XNM                                                ANSN2382
      IF(AMIN1(E5,E6) .GT. 0.0)GO TO 6007                             ANSN2383
      IF(IFLU-1)6002,6007,6003                                        ANSN2384
C  * * * * * * * * * *   STEP FUNCTION MODEL                          *ANSN2385
 6002 E3=(XNRI + DSLM*XNM + DAIM*PHIM)/(E1 + DSKM + DBIM)             ANSN2386
      E5=E3                                                           ANSN2387
      E6=E3                                                           ANSN2388
      GO TO 6007                                                      ANSN2389
C  * * * * * * * * * *   WEIGHTED DIFFERENCE MODEL                    *ANSN2390
C  * * * * * * * * * *   COMPUTE WEIGHTED SPACE TERMS                *ANSN2391
 6003 IF(XNM.LE.0.0)GO TO 7001                                        ANSN2392
 7005 XNUM=E1 + DBIM + DBIM                                           ANSN2393
      XDEN=XNUM - DSLM - (DAIM*PHIM + XNRI)*0.9/XNM                   ANSN2394
      IF(XDEN+XDEN .LE. XNUM) GO TO 7001                              ANSN2395
      IF(XDEN-XNUM)7006,7003,7001                                     ANSN239
 7006 BBB=XNUM/XDEN                                                   ANSN2397
 7004 XD1=DSLM - DSKM + BBB*DSKM                                      ANSN2398
      GO TO 7002                                                      ANSN2399
 7003 XD1=DSLM                                                        ANSN2400
      BBB=1.0                                                         ANSN2401
      GO TO 7002                                                      ANSN2402
 7001 XD1=DSKM2                                                       ANSN2403
      BBB=2.0                                                         ANSN2404
C  * * * * * * * * * *   COMPUTE WEIGHTED ANGLE TERMS                *ANSN2405
 7002 IF(PHIM.LE.0.0)GO TO 7011                                       ANSN2406
 7015 XNUM=E1 + DSKM + DSKM                                           ANSN2407
      XDEN=XNUM - DAIM - (DSLM*XNM + XNRI)*0.9/PHIM                   ANSN2408
      IF(XDEN+XDEN .LE. XNUM) GO TO 7011                              ANSN2409
      IF(XDEN-XNUM)7016,7013,7011                                     ANSN2410
 7016 CCC=XNUM/XDEN                                                   ANSN2411
 7014 XD2=DAIM - DBIM + CCC*DBIM                                      ANSN2412
      GO TO 6004                                                      ANSN2413
```

Figure B1 (contd.)

```
 7013 XD2=DAIM                                                          ANSN2414
      CCC=1.0                                                           ANSN2415
      GO TO 6004                                                        ANSN2416
 7011 XD2=DBIM2                                                         ANSN2417
      CCC=2.0                                                           ANSN2418
 6004 E3=(XNRI + XD1*XNM + XD2*PHIM)/(E1 + XD1 + XD2)                   ANSN2419
      E5=CCC*(E3-PHIM) + PHIM                                           ANSN2420
      E6=BBB*(E3-XNM)  + XNM                                            ANSN2421
      GO TO 6007                                                        ANSN2422
C  * * * * * * * * * *   INITIAL DIRECTION FLUX CALCULATION  -  W.EQ.0 *ANSN2423
C  * * * * * * * * * *   CR SLAB FLUX CALCULATION  -  IGE.EQ.1         *ANSN2424
 6006 IF(IFLU-2)6011,6012,6013                                          ANSN2425
C  * * * * * * * * * *   LINEAR MODEL                                  *ANSN2426
 6011 E3=(XNRI + DSKM2*XNM)/(E1 + DSKM2)                                ANSN2427
      E5=E3                                                             ANSN2428
      E6=E3 + E3 - XNM                                                  ANSN2429
      IF(E6.GT.0.0)GO TO 6007                                           ANSN2430
      IF(IFLU-1)6012,6007,6013                                          ANSN2431
C  * * * * * * * * * *   STEP FUNCTION MODEL                           *ANSN2432
 6012 E3=(XNRI + DSLM*XNM)/(E1 + DSLM)                                  ANSN2433
      E5=E3                                                             ANSN2434
      E6=E3                                                             ANSN2435
      GO TO 6007                                                        ANSN2436
C  * * * * * * * * * *   WEIGHTED DIFFERENCE MODEL                     *ANSN2437
C  * * * * * * * * * *   COMPUTE WEIGHTED SPACE TERMS                 *ANSN2438
 6013 IF(XNM.LE.0.0)GO TO 7021                                          ANSN2439
 7025 DBIM2=DSLM - DSKM                                                 ANSN2440
      XNUM=E1 + DBIM2                                                   ANSN2441
      XDEN=XNUM - DSLM - XNRI*0.9/XNM                                   ANSN2442
      IF(XDEN+XDEN .LE. XNUM) GO TO 7021                                ANSN2443
      IF(XDEN-XNUM)7026,7023,7021                                       ANSN2444
 7026 BBB=XNUM/XDEN                                                     ANSN2445
 7024 XD1=DBIM2 + BBB*DSKM                                              ANSN2446
      GO TO 7022                                                        ANSN2447
 7023 XD1=DSLM                                                          ANSN2448
      BBB=1.0                                                           ANSN2449
      GO TO 7022                                                        ANSN2450
 7021 XD1=DSKM2                                                         ANSN2451
      BBB=2.0                                                           ANSN2452
 7022 E3=(XNRI + XD1*XNM)/(E1 + XD1)                                    ANSN2453
      E5=E3                                                             ANSN2454
      E6=BBB*(E3-XNM) + XNM                                             ANSN2455
 6007 CONTINUE                                                          ANSN2456
C SPUTTERING CHANGE - SAVE CSD FOR CURRENT GROUP
 6008   E3SAV1(I,M)=E3*SIGSD(I1G)
      E3=E3*WM                                                          ANSN2457
      XNE(I)=E5                                                         ANSN2458
      XNM=E6                                                            ANSN2459
C   *** COMPUTE TOTAL FLUX (SCALAR)                                     ANSN2460
      XNN(I)=XNN(I) + E3                                                ANSN2461
      IF(ISCT.EQ.0)GO TO 33                                             ANSN2462
C   *** COMPUTE P-L CURRENTS                                            ANSN2463
```

Figure B1 (contd.)

```
      IN=I                                                              ANSN2464
      MN=M                                                              ANSN24 5
      DO 32 M=1,JT                                                      ANSN2466
      XNA(IN)=XNA(IN) + PNC(MN)*E3                                      ANSN24 7
      IN=IN + IM                                                        ANSN2468
   32 MN=MN + MM                                                        ANSN2469
   33 IF(DSNM.LT.0.0)GO TO 25                                           ANSN2470
C    *** ADD SHELL SOURCE TO FLUX WHEN MU.GT.0                         ANSN2471
      IF(IPM-1)25,34,35                                                 ANSN2472
   35 IGMIMI=IGMIP - [illegible]                                        ANSN2473
      XNM=XNM + PA(IGMIMI)                                              ANSN2474
      GO TO 25                                                          ANSN2475
   34 IF(IPP.EQ.I)XNM=XNM +PAIGMM                                       ANSN2476
   25 CONTINUE                                                          ANSN2477
C    *** SAVE BOUNDARY FLUX                                            ANSN2478
      IGMPMM=IGMPM + M                                                  ANSN2479
      B(IGMPMM)=E6                                                      ANSN2480
C    *** SAVE LAST ANGULAR FLUX                                        ANSN2481
      IPMML=IPMM + IABS(IOO-IP)                                         ANSN24 2
      XND(IPMM[illegible])=E6                                           ANSN2483
    5 CONTINUE                                                          ANSN2484
  845 E1=0.0                                                            ANSN2485
      E2=0.0                                                            ANSN2486
      IF(IBL.EQ.1)GO TO 3337                                            ANSN2487
      J=1                                                               ANSN2488
      DO 37 M=1,MM                                                      ANSN2489
      E1=E1 + WD(M)*XND(I)                                              ANSN2490
   37 I=I + IP                                                          ANSN2491
 3337 J=IP                                                              ANSN2492
      DO 337M=1,MM                                                      ANSN2493
      E2=E2 + WD(M)*XND(J)                                              ANSN2494
  337 J=J + IP                                                          ANSN2495
C    *** COMPUTE LEAKAGE                                               ANSN2496
      RL=E2*AA(IP)                                                      ANSN2497
      XLL=E1*AA(1)                                                      ANSN2498
      XNLGG=RL - XLL                                                    ANSN2499
  833 IIC=IIC+1                                                         ANSN2500
      E1=0.0                                                            ANSN2501
      ESC=0.0                                                           ANSN2502
      ESM=0.0                                                           ANSN2503
      EAM=0.0                                                           ANSN2504
C    *** COMPUTE CONVERGENCE NUMBERS                                   ANSN2505
      IGMPII=IGMPI + 1                                                  ANSN2506
      DO 38 I=1,IM                                                      ANSN2507
      E4=XNN(I) - XN(IGMPII)                                            ANSN2508
      IGMPII=IGMPII + 1                                                 ANSN2509
      E2=CS(I)*E4                                                       ANSN2510
      E1=E1+E2                                                          ANSN2511
      ESM=ESM + ABS(E2)                                                 ANSN2512
      EAM=EAM + ABS(E4)*(CT(I)-CS(I))                                   ANSN2513
      E4=ABS(E4 /XNN(I))                                                ANSN2514
      IF(E4.LT.ESC)GO TO 38                                             ANSN2515
```

Figure B1 (contd.)

```
      IFDMX=I                                                          ANSN2516
      ESC=E4                                                           ANSN2517
   38 CONTINUE                                                         ANSN2518
      ESM=AMAX1(ESM,EAM)                                               ANSN2519
C     *** COMPUTE INNER ITERATION SCALE FACTOR                         ANSN2520
      E1=ZZ1/(ZZ1-E1)                                                  ANSN2521
      IF(E1.LE.0.0.OR.E1.GT.10.0)E1=10.0                               ANSN2522
      IF(ICMC.EQ.1)GO TO 124                                           ANSN2523
C     *** COMPUTE SPACE DEPENDENT SCALE FACTORS                        ANSN2524
C     *** COMPUTE LEAKAGE TERMS AND SHELL SOURCE CONTRIBUTION TO G     ANSN2525
      DO 100 I=1,ICMC                                                  ANSN2526
      DC(IM +I)=0.0                                                    ANSN2527
      XNR(I)=0.0                                                       ANSN2528
  100 XNE(I)=0.0                                                       ANSN2529
      IGMPII=IGMPI + 1                                                 ANSN2530
      DO 101 I=1,IM                                                    ANSN2531
      II=NO(I)                                                         ANSN2532
      XNE(II)=XNE(II) + SR(I)                                          ANSN2533
C SPUTTERING CHANGE - PRESERVE BALANCE FOR CONVERGENCE TESTS
        DO 888 M=1,MM
  888   XNE(II)=XNE(II)+E3SAV2(I,M)*V(I)*W(M)
      XNR(II)=XNR(II) + CS(I)*(XN(IGMPII)-XNN(I))                      ANSN2534
      DC(IM +II)=DC(IM +II) + CS(I)*XNN(I)                             ANSN2535
      SP=0.0                                                           ANSN2536
      SM=0.0                                                           ANSN2537
      IF(IPM-1)108,102,103                                             ANSN2538
  102 IF(IPP.NE.I)GO TO 108                                            ANSN2539
      J=1                                                              ANSN2540
      GO TO 104                                                        ANSN2541
  103 J=I                                                              ANSN2542
  104 IGMIJ=IGMI + J*MM + 1                                            ANSN2543
      DO 1011 M=1,MM                                                   ANSN2544
      E3=PA(IGMIJ)*ABS(WD(M))                                          ANSN2545
      IGMIJ=IGMIJ+1                                                    ANSN2546
      IF(WD(M).LT.0.0)GO TO 105                                        ANSN2547
      SP=SP + E3                                                       ANSN2548
      GO TO 1011                                                       ANSN2549
  105 SM=SM + E3                                                       ANSN2550
 1011 CONTINUE                                                         ANSN2551
      XNE(II)=XNE(II) + SM*AA(I+1)                                     ANSN2552
      IF(I+1 .EQ. NO(IM+II))GO TO 106                                  ANSN2553
      XNE(II)=XNE(II) + SP*AA(I+1)                                     ANSN2554
      GO TO 108                                                        ANSN2555
  106 IF(II.NE.ICMC)XNE(II+1)=XNE(II+1) + SP*AA(I+1)                   ANSN2556
  108 IGMPII=IGMPII+1                                                  ANSN2557
  101 CONTINUE                                                         ANSN2558
      II1=1                                                            ANSN2559
      DO 109 I=1,ICMC                                                  ANSN2560
      DC(IM +I)=DC(IM +I)*0.25                                         ANSN2561
      II2=NO(IM+I)                                                     ANSN2562
      II1S=II1                                                         ANSN2563
      II2S=II2                                                         ANSN2564
```

Figure B1 (contd.)

```
      XLM=0.0                                                          ANSN2565
      XLP=0.0                                                          ANSN2566
      DO 115 M=1,MM                                                    ANSN2567
      IF(WD(M).GT.0.0)GO TO 116                                        ANSN2568
      XLP=XLP + XND(II2)*ABS(WD(M))                                    ANSN2569
      GO TO 117                                                        ANSN2570
  116 XLM=XLM + XND(II1)*WD(M)                                         ANSN2571
  117 II1=II1 + IP                                                     ANSN2572
  115 II2=II2 + IP                                                     ANSN2573
      II1=II2S                                                         ANSN2574
      ST(I)=-XLP*AA(II2S)                                              ANSN2575
  109 SAT(I)=-XLM*AA(II1S)                                             ANSN2576
      FLUXP=DC(IM +1)                                                  ANSN2577
      DO 118 I=1,ICMC                                                  ANSN257
      FLUXM=FLUXP                                                      ANSN2579
      IF(I.EQ.ICMC)GO TO 8001                                          ANSN2580
      FLUXP=AMAX1(DC(IM +I),DC(IM +I+1))                               ANSN2581
      GO TO 8002                                                       ANSN2582
 8001 FLUXP=DC(IM +I)                                                  ANSN2583
 8002 ST(I)=ST(I)-FLUXP                                                ANSN2584
      SAT(I)=SAT(I)-FLUXM                                              ANSN2585
  118 XNR(I)=XNR(I) + XNE(I) - ST(I) - SAT(I)                          ANSN2586
C     ***                                                             ANSN2587
C     *** SOLVE TRI-DIAGONAL MATRIX                                   ANSN25
C     ***                                                             ANSN2589
C     *** COMPUTE A(1) AND A(IP)                                      ANSN2590
      XNR(1)=XNR(1) + SAT(1)                                           ANSN2591
      XNR(ICMC)=XNR(ICMC) + ST(ICMC)                                   ANSN2592
C     *** REMOVE B TERMS FROM MATRIX                                  ANSN2593
      J=1                                                              ANSN2594
      DO 111 I=2,ICMC                                                  ANSN2595
      COEF=SAT(I)/XNR(J)                                               ANSN2596
      XNR(I)=XNR(I) - COEF*ST(J)                                       ANSN2597
      XNE(I)=XNE(I) - COEF*XNE(J)                                      ANSN2598
  111 J=J+1                                                            ANSN2599
C     *** SOLVE FOR F(I)                                              ANSN2600
      I=ICMC                                                           ANSN2601
      DO 112 J=2,ICMC                                                  ANSN2602
      XNE(I)=XNE(I)/XNR(I)                                             ANSN2603
      IF(XNE(I).LT.0.0)GO TO 124                                       ANSN2604
      I=I-1                                                            ANSN2605
  112 XNE(I)=XNE(I) - XNE(I+1)*ST(I)                                   ANSN2606
      XNE(1)=XNE(1)/XNR(1)                                             ANSN2607
      IF(XNE(1).LT.0.0)GO TO 124                                       ANSN2608
C     ***                                                             ANSN2609
C     *** MATRIX SOLVED                                               ANSN2610
C     ***                                                             ANSN2611
      I=IM                                                             ANSN2612
      DO 119 J=1,IM                                                    ANSN2613
      II=NO(I)                                                         ANSN2614
      XNE(I)=XNE(II)                                                   ANSN2615
  119 I=I-1                                                            ANSN2616
```

Figure B1 (contd.)

Figure A1 (contd.)

```
      DO 114 I=1,IM                                                    ANSN2617
  114 SAT(I)=0.0                                                       ANSN2618
      GO TO 125                                                        ANSN2619
C   *** APPLY SPACE DEPENDENT SCALE FACTORS                            ANSN2620
  124 DO 123 I=1,IM                                                    ANSN2621
  123 XNE(I)=E1                                                        ANSN2622
  125 DO 39 I=1,IM                                                     ANSN2623
      IF(ABS(1.0 - XNE(I)).LT.SCLMX)GO TO 39                           ANSN2624
      SCLMX=ABS(1.0 - XNE(I))                                          ANSN2625
      SCLXX=XNE(I)                                                     ANSN2626
      ISMX=I                                                           ANSN2627
   39 XNN(I)=XNN(I)*XNE(I)                                             ANSN2628
  122 IF(ISCT.EQ.0)GO TO 3399                                          ANSN2629
      DO 339 I=1,IM                                                    ANSN2630
      IN=I                                                             ANSN2631
      DO 40 N=1,JT                                                     ANSN2632
      XNA(IN)=XNA(IN)*XNE(I)                                           ANSN2633
   40 IN=IN + IM                                                       ANSN2634
  339 CONTINUE                                                         ANSN2635
 3399 E1=XNE(1)                                                        ANSN2636
      E2=XNE(IM)                                                       ANSN2637
      IGMPMM=IGMPM+1                                                   ANSN2638
      DO 41 M=1,MM                                                     ANSN2639
      IF(WD(M).GT.0.0)GO TO 441                                        ANSN2640
      B(IGMPMM)=B(IGMPMM)*E1                                           ANSN2641
      GO TO 41                                                         ANSN2642
  441 B(IGMPMM)=B(IGMPMM)*E2                                           ANSN2643
   41 IGMPMM=IGMPMM + 1                                                ANSN2644
C   *** TEST CONVERGENCE                                               ANSN2645
      IF((XITR.EQ.0 .AND. IBR.EQ.0) .OR. IIC.GE.IIM)GO TO 42           ANSN2646
      IF(ESM.GT.EPS .AND. ESC.GT.EPS  .OR.  XLAL.GT.0.0 .AND.          ANSN2647
     1 ESC.GT.XLAL)GO TO 832                                           ANSN2648
   42 K=1                                                              ANSN2649
      DO 43 M=1,MM                                                     ANSN2650
      WDM=WD(M)                                                        ANSN2651
      J=K+1                                                            ANSN2652
      DO 443 I=1,IM                                                    ANSN2653
      XNEI=XNE(I)                                                      ANSN2654
      IF(WDM.LT.0.0 .OR. I.EQ.1)XND(K)=XND(K)*XNEI                     ANSN2655
      IF(WDM.GT.0.0 .OR. I.EQ.IM)XND(J)=XND(J)*XNEI                    ANSN2656
      K=K+1                                                            ANSN2657
C SPUTTERING CHANGE - SAVE CSD FOR NEXT GROUP
        E3SAV2(I,M)=E3SAV1(I,M)
  443 J=J+1                                                            ANSN2658
      K=J                                                              ANSN2659
   43 CONTINUE                                                         ANSN2660
      XNLGG=RL*E2 - XLL*E1                                             ANSN2661
      IF(ICVT.NE.1)GO TO 999                                           ANSN2662
      IF(IIG.EQ.1)WRITE(NOU,60)                                        ANSN2663
      WRITE(NOU,70) IIG,IIC,IFDMX,ESC,ISMX,SCLXX,ICMC                  ANSN2664
   60 FORMAT(76X,' GROUP INNER   MFD    MAX. FLUX   MSF   MAX. SCALE COAANSN2665
     *RSE'/83X,'ITERS  INT.   DIFFERENCE  INT.     FACTOR    MESH')     ANSN2666
```

Figure B1 (contd.)

```
   70 FORMAT(76X,3I6,1PE13.5,I6,E13.5,I6)                                  ANSN2667
  999 RETURN                                                               ANSN2668
      END                                                                  ANSN2669
      SUBROUTINE SUMARY(XKI,FD,XN,  VE,W,DSN,MA,MZ,CRX,Q,PA,DF,I2,CA,CF,ANSN3072
     1CT,CS,V,AA,TAB,WD,XND,FXS,FIS,SNN,SFS,OTS,APS,FRT,TXN,DEN,RFL,RCT,ANSN3073
     2RLK,XLL,XLK,XBB,RXN,IG1,I1,IH1,IGM1,M1,IM1)                          ANSN3074
C   *** SUMARY PRINTS ANGULAR FLUXES AND COMPUTES SUMMARY TABLES           ANSN3075
C SPUTTERING CHANGE - SETUP ARRAYS, COMMON/SPUT/
      REAL*8 EBT,UO
      DIMENSION E(50,2),R(50,33),ETA(16,2),WT(16)
      COMMON/SPUT/SIGSD(50),EBT(51),UO,NI
     DIMENSION XKI(1),FD(1),XN(1,1),     VE(1),W(1),DSN(1),MA(1),MZ(1)ANSN3076
     1,CRX(IH1,IGM1,1),Q(I1,1),PA(M1,IM1,1),DF(1),CA(1),CF(1),CT(1),    ANSN3077
     2 CS(1),V(1),AA(1),TAB(1),WD(1),XND(I2,1),FXS(IG1,1),FIS(IG1,1),   ANSN3078
     3 SNN(IG1,1),SFS(IG1,1),OTS(IG1,1),APS(IG1,1),XLK(IG1,1),XBB(IG1,1)ANSN3079
     4,RXN(IG1,1),RFL(IG1,1),RCT(IG1,1),RLK(IG1,1),XLL(IG1,1),FRT(IG1,1)ANSN3080
     5,TXN(IG1,1),DEN(IG1,1)                                            ANSN3081
     COMMON /BULKPU/                                                    ANSN3082
     $       D(1),LIM1,LXM1,LM1,LXN,LR,LVE,LW,LDSN,LMA,LMZ,LMB,LMC,LXMD,ANSN3083
     1LMTT,LCRX,LFIX,LFLT,LQ,LPA,LJ5,LRM,LDF,LJ3,LJ4,LIGT,LART,LALFT,   ANSN3084
     2LFGP,LFGG,LEND,LV,LAA,LWD,LMF,LPNC,LXJ,LCH,LCA,LCF,LCT,LCS,LTAB,  ANSN3085
     3LXND,LSA,LSAT,LRAV,LRA,LXNN,LXNE,LXNR,LXNA,LSR,LST,LQG,LFG,LSG,   ANSN3086
     4LXKF,LXNI,LXNO,LT3,LT5,LDA,LDB,LDC,LDS,LB,IGMP,IGMM,IIGG,NERR,IMJTANSN3087
     5,IHG,IMP,MP,NDS,NUS,SDG,SCG,AG,XNLGG,XNLG,SNG,ALA,ASR,EAM,EPG,EQ, ANSN3088
     6E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,E11,E12,E13,E14,E15,E16,E17,E18,E19ANSN3089
     7,E20,ESC,ESM,EVP,EVPP,FTP,IC,ICVT,IGP,IG,IHP,IIC,IIG,IP,IZT,IO1,  ANSN3090
     8IO2,IO3,IO4,IO5,IO6,IO7,IO8,IO9,IOO,JT,LC,MG,MI,ML,MM,NFN,XITR,   ANSN3091
     9XLAP,XLAPP,XLAR,XLA,XNIO,XNII,ZZ1,ZZ2,ZZ3,XNB,XKEP,XKIP,IH,I,K,L, ANSN3092
     AM,J,N,NN,ISV,                                                     ANSN3093
     EID,ITH,ISCT,ISN,IGE,IBL,IBR,IZM,IM,IEVT,IGM,IHT,IHS,IHM,MS,MCR,MTPANSN3094
     C,MT,IDFM,IPVT,IQM,IPM,IPP,JIM,ID1,ID2,ID3,ID4,ICM,IDAT1,IDAT2,IPG,ANSN3095
     DIFLU,IFN,IPRT,IXTR,                                               ANSN3096
     EEV,EVM,EPS,BF,DY,DZ,DFM1,XNF,PV,RYF,XLAI,XLAH,EQL,XNPM,           ANSN3097
     FT(12),NIN,NOU,NT1,NT2,NT3,NT4,NT5,NT6,NT7                         ANSN3098
C SPUTTERING CHANGE - CALCULATE ANGULAR RESPONSE FUNCTION, R
      NT=ICM-NI
      JSN=ISN
      WRITE(6,2000)NI,NT,JSN,UO
 2000 FORMAT(1H1,I5,' ICN GROUPS;',I5,'TARGET ATOM GROUPS;  S',I2,
     1  '  UO=',D15.5)
      JSN=JSN/2
      NTP1=NT+1
      JSN1=JSN+1
      DO 105 I=1,JSN
  105 WT(I)=W(I+1)
      WRITE(6,2005)
 2005 FORMAT('0  TARGET ATOM GROUP BOUNDARIES AND SN WEIGHTS')
      WRITE(6,2010)(EBT(I),I=1,NTP1)
      WRITE(6,2010)(WT(I),I=1,JSN)
 2010 FORMAT(1H ,7E15.5)
      DO 110 JG=1,NT
      E(JG,1)=EBT(JG+1)
```

Figure B1 (contd.)

```
110     E(JG,2)=EET(JG)
        ETA(1,1)=0.0
        ETA(1,2)=2.*WT(1)
        DO 120 IA=2,JSN
        ETA(IA,1)=ETA(IA-1,2)
120     ETA(IA,2)=ETA(IA,1)+2.*WT(IA)
        JSN2=2*JSN+1
        NTOT=NI+NT
        DO 130 JG=1,NTOT
        DO 130 IA=1,JSN2
130     R(JG,IA)=0.0
        DO 140 JG=1,NT
        IGG=NI+JG
        DO 140 IA=1,JSN
        IAA=JSN1-IA+1
        EA=E(JG,1)*ETA(IA,1)**2
        EB=E(JG,2)*ETA(IA,1)**2
        EC=E(JG,1)*ETA(IA,2)**2
        ED=E(JG,2)*ETA(IA,2)**2
        IF(EA.GT.UO) GO TO 101
        IF(ED.LT.UO) GO TO 106
        IF(EC.GT.UO) GO TO 102
        ETAL=ETA(IA,1)**2
        IF(EB.LT.UO) ETAL=UO/E(JG,2)
        R(IGG,IAA)=(ETA(IA,2)**2+ETAL)/2./UO-1./E(JG,2)
        R(IGG,IAA)=R(IGG,IAA)/(1./E(JG,1)-1./E(JG,2))
        IF(EB.LT.UO) R(IGG,IAA)=R(IGG,IAA)*(ETA(IA,2)**2-ETAL)/
     1 (ETA(IA,2)**2-ETA(IA,1)**2)
        GO TO 140
101     R(IGG,IAA)=1.0
        GO TO 140
102     EU=E(JG,2)
        IF(EB.GT.UO) EU=UO/ETA(IA,1)**2
        R(IGG,IAA)=UO*(1./E(JG,1)+1./EU)/2.-ETA(IA,1)**2
        IF(EB.GT.UO) R(IGG,IAA)=R(IGG,IAA)*(1./E(JG,1)-1./EU)/
     1 (1./E(JG,1)-1./E(JG,2))
        R(IGG,IAA)=1.-R(IGG,IAA)/(ETA(IA,2)**2-ETA(IA,1)**2)
        GO TO 140
106     R(IGG,IAA)=0.0
140     CONTINUE
        WRITE(6,2015)
2015    FORMAT(///'  ANGULAR LEAKAGE RESPONSE FOR SPUTTERING PROBLEM'/)
        DO 170 JG=1,NTOT
        DO 169 IA=2,JSN1
        RTEMP=R(JG,IA)
        IATP=JSN2-IA+2
        R(JG,IA)=R(JG,IATP)
169     R(JG,IATP)=RTEMP
        WRITE(6,2020) JG,JSN2
170     WRITE(6,2010) (R(JG,IA),IA=1,JSN2)
2020    FORMAT('                    R(',I2,',IA),IA=1,',I2)
     DO 1 IIC=1,IGM                                                    ANSN3099
```

Figure B1 (contd.)

```
      IIGG=IIG                                                          ANSN3100
      CALL WANDR1 ( 3 ,XND,IP*MM,2)
      IF(ID1.EQ.C .OR. ID1.EQ.2)GO TO 2                                 ANSN3102
C     *** PRINT ANGULAR FLUX                                            ANSN3103
      WRITE (NOU,10)IIG                                                 ANSN3104
   10 FORMAT(37H1 FLUX BY ANGLE AND POINT FOR GROUP  I3)                ANSN3105
      CALL WOT(XND,MM,IP,1,'PNT.','ANGL',0,NOU)
    2 IF(IDAT1.EQ.0)GO TO 3                                             ANSN3107
      IIGG=1                                                            ANSN3108
      CALL WANDR1 (NT3,CRX,IHP*MT,2)                                    03839000
      IO1=MP*IMP                                                        ANSN3110
      IF(IQM+IPM.NE.0)  CALL WANDR1 (NT3,0,IO1,2)
C     *** COMPUTE SUMMARY TABLES                                        ANSN3112
    3 IH=IHM                                                            ANSN3113
    4 K=IIG+IES-IH                                                      ANSN3114
      IF(K.LE.0)GO TO 5                                                 ANSN3115
      IF(K.GT.IGM)GO TO 6                                               ANSN3116
      IF(IDAT1.NE.2)GO TO 7                                             ANSN3117
      K=1                                                               ANSN3118
      CALL WANDR1 (NT1,XN,IM,2)                                         03453000
      IF(ISCT.GT.0)READ (NT1)                                           ANSN3120
    7 DO 8 I=1,IM                                                       ANSN3121
      J=MA(I)                                                           ANSN3122
      L=IABS(MZ(J))                                                     ANSN3123
      J=MINO(J,IZM)                                                     ANSN3124
      E4=V(I)                                                           ANSN3125
      IF(IDFM.GT.0)E4=E4*DF(I)                                          ANSN3126
      IF(IH.NE.IHS)GO TO 39                                             ANSN3127
      CS(I)=XN(I,K)                                                     ANSN3128
      E1=CRX(IHT,IIGG,L)                                                ANSN3129
      E2=CRX(IHT-2,IIGG,L)                                              ANSN3130
      E3=0.0                                                            ANSN3131
      IF(IDFM.EQ.0)GO TO 11                                             ANSN3132
      E1=E1*DF(I)                                                       ANSN3133
      E2=E2*DF(I)                                                       ANSN3134
C     *** TRANSVERSE BUCKLING CORRECTION                                ANSN3135
   11 IF(DY.GT.0.0) E3=3.289868*E1/(DY*E1 +BF)**2                       ANSN3136
      IF(DZ.GT.0.0) E3=E3 + 3.289868*E1/(DZ*E1 + BF)**2                 ANSN3137
      CT(I)=(E1 + TAB(IIG) + E3)*CS(I) *V(I)                            ANSN3138
      CA(I)=(E2 + TAB(IIG) + E3)*CS(I) *V(I)                            ANSN3139
      GO TO 8                                                           ANSN3140
C     *** FXS - FIXED SOURCE                                            ANSN3141
C     *** FIS - FISSION SOURCE                                          ANSN3142
C     *** SNN - SCATTER GAINS                                           ANSN3143
C     *** SFS - SELF SCATTER                                            ANSN3144
C     *** OTS - SCATTER LOSSES                                          ANSN3145
C     *** APS - ABSORPTIONS                                             ANSN3146
C     *** XLK - NET LEAKAGE                                             ANSN3147
C     *** XBB - NEUTRON BALANCE  (GAINS/LOSSES)                         ANSN3148
C     *** RXN - FLUX AT RIGHT BOUNDARY                                  ANSN3149
C     *** RFL - J+ AT RIGHT BOUNDARY                                    ANSN3150
C     *** RCT - J-NET AT RIGHT BOUNDARY                                 ANSN3151
```

Figure B1 (contd.)

```
C     *** RLK - RIGHT LEAKAGE                                          ANSN3152
C     *** XLL - LEFT LEAKAGE                                           ANSN3153
C     *** FRT - FISSION RATE                                           ANSN3154
C     *** TXN - TOTAL FLUX                                             ANSN3155
C     *** DEN - DENSITY                                                ANSN3156
   39 SNN(IIG,J)=SNN(IIG,J) + CRX(IH,IIGG,L)*XN(I,K)*EA               ANSN3157
    8 CONTINUE                                                        ANSN3158
    5 IH=IH-1                                                         ANSN3159
      IF(IH.GT.IHT)GO TO 4                                            ANSN3160
    6 IF(IDAT1.NE.2)GO TO 35                                          ANSN3161
      IF(NDS.LT.IIG .AND. IIG.NE.IGM)GO TO 43                         ANSN3162
      REWIND NT1                                                      ANSN3163
      GO TO 35                                                        ANSN3164
   43 K=NDS + MINO(NUS,IGM-IIG)                                       ANSN3165
      IF(ISCT.GT.0)K=K+N                                              ANSN3166
      DO 36 I=1,K                                                     ANSN3167
   36 BACKSPACE NT1                                                   ANSN3168
   35 E1=EV                                                           ANSN3169
      IF(IEVT.NE.1)E1=1.0                                             ANSN3170
      IGI=(IIGG-1)*MP*IMP                                             ANSN3171
      DO 9 I=1,IM                                                     ANSN3172
      J=MA(I)                                                         ANSN3173
      K=IABS(MZ(J))                                                   ANSN3174
      J=MINO(J,IZM)                                                   ANSN3175
      EA=V(I)                                                         ANSN3176
      IF(IGFM.GT.0)EA=EA*LF(I)                                        ANSN3177
      IF(ITH.EQ.0)GO TO 41                                            ANSN3178
      FIS(IIG,J)=FIS(IIG,J) + CRX(IHT-1,IIGG,K)*FD(1)*EA              ANSN3179
      GO TO 42                                                        ANSN3180
   41 FIS(IIG,J)=FIS(IIG,J) + XKI(IIG)*FD(I)*V(I)/E1                  ANSN3181
   42 SFS(IIG,J)=SFS(IIG,J) + CS(I)*CRX(IHS,IIGG,K) *EA               ANSN3182
      OTS(IIG,J)=OTS(IIG,J) + CT(I)                                   ANSN3183
      APS(IIG,J)=APS(IIG,J) + CA(I)                                   ANSN3184
      TXN(IIG,J)=TXN(IIG,J) + CS(I)*V(I)                              ANSN3185
      IF(IQM.NE.0)FXS(IIG,J)=FXS(IIG,J) + Q(IGI+I,1)*V(1)             ANSN3186
    9 FRT(IIG,J)=FRT(IIG,J) + CS(I)*CRX(IHT-1,IIGG,K)*EA              ANSN3187
      DO 12 J=1,IZM                                                   ANSN3188
      DEN(IIG,J)=TXN(IIG,J)/VE(IIG)                                   ANSN3189
   12 OTS(IIG,J)=OTS(IIG,J) - APS(IIG,J) - SFS(IIG,J)                 ANSN3190
      IF(IPM.EQ.0)GO TO 13                                            ANSN3191
      J=MINO(IZM,MA(IPP))                                             ANSN3192
      JJ=J                                                            ANSN3193
      IF(IPP.NE.IM)JJ=MINO(IZM,MA(IPP+1))                             ANSN3194
      DO 17 I=1,IM                                                    ANSN3195
      IF(IPM.EQ.1)GO TO 1007                                          ANSN3196
      J=MINO(IZM,MA(I))                                               ANSN3197
      JJ=J                                                            ANSN3198
      IF(I.NE.IM)JJ=MINO(IZM,MA(I+1))                                 ANSN3199
 1007 E3=0.0                                                          ANSN3200
      EA=0.0                                                          ANSN3201
      DO 16 M=1,MM                                                    ANSN3202
      E2=PA(M,I,IIGG)*ABS(WD(M))                                      ANSN3203
```

Figure B1 (contd.)

```
      IF(WD(M).LT.0.0)GO TO 1005                                        ANSN3204
      E3=E3 + E2                                                        ANSN3205
      GO TO 16                                                          ANSN3206
 1005 E4=E4 + E2                                                        ANSN3207
   16 CONTINUE                                                          ANSN3208
      IF(IPM.EQ.1)GO TO 15                                              ANSN3209
      IF(I.EQ.IM)GO TO 17                                               ANSN3210
      FXS(IIG,JJ)=FXS(IIG,JJ) + E3*AA(I+1)                             ANSN3211
   17 FXS(IIG,J)=FXS(IIG,J) + E4*AA(I+1)                                ANSN3212
      GO TO 13                                                          ANSN3213
   15 IF(IPP.EQ.IM)GO TO 1008                                           ANSN3214
      FXS(IIG,JJ)=FXS(IIG,JJ) + E3*AA(IPP+1)                           ANSN3215
 1008 FXS(IIG,J)=FXS(IIG,J) + E4*AA(IPP+1)                              ANSN3216
   13 DO 18 J=1,IZP                                                     ANSN3217
      IF(J.NE.IZP .AND. IZM.NE.1)GO TO 24                               ANSN3218
      I01=1                                                             ANSN3219
      I02=IM                                                            ANSN3220
      GO TO 25                                                          ANSN3221
   24 DO 19 I01=1,IM                                                    ANSN3222
      IF(J.EQ.MA(I01))GO TO 21                                          ANSN3223
   19 CONTINUE                                                          ANSN3224
   21 I02=IP                                                            ANSN3225
   22 I02=IC2-1                                                         ANSN3226
      IF(J.NE.MA(I02))GO TO 22                                          ANSN3227
   25 DO 23 M=1,MM                                                      ANSN3228
      RXN(IIG,J)=RXN(IIG,J) + XND(I02+1,M)*W(M)                         ANSN3229
      IF(WD(M).GT.0.0)RFL(IIG,J)=RFL(IIG,J) + XND(I02+1,M)*WD(M)        ANSN3230
C SPUTTERING CHANGE - FOLD R WITH RIGHT LEAKAGE
        RCT(IIG,J)=RCT(IIG,J)+XND(I02+1,M)*WD(M)*R(IIG,M)
   23 XLL(IIG,J)=XLL(IIG,J)+XND(I01,M)*WD(M)                            ANSN3232
      RLK(IIG,J)=RCT(IIG,J)*AA(I02+1)                                   ANSN3233
      XLL(IIG,J)=XLL(IIG,J)*AA(I01)                                     ANSN3234
      XLK(IIG,J)=RLK(IIG,J) - XLL(IIG,J)                                ANSN3235
   18 CONTINUE                                                          ANSN3236
    1 CONTINUE                                                          ANSN3237
      REWIND 3
      IF(IDAT1.NE.0)REWIND NT3                                          ANSN3239
   37 DO 34 K=1,16                                                      ANSN3240
      L=(K-1)*IZP                                                       ANSN3241
      M=L+IZP                                                           ANSN3242
      DO 26 J=1,IZP                                                     ANSN3243
      N=L+J                                                             ANSN3244
      DO 27 IIG=1,IGM                                                   ANSN3245
      E1=FXS(IIG,N)                                                     ANSN3246
      IF(J.EQ.IZP)GO TO 28                                              ANSN3247
      FXS(IGP,N)=FXS(IGP,N) + E1                                        ANSN3248
      IF(K.GT.9)GO TO 27                                                ANSN3249
      FXS(IIG,M)=FXS(IIG,M) + E1                                        ANSN3250
      GO TO 27                                                          ANSN3251
   28 FXS(IGP,M)=FXS(IGP,M) + E1                                        ANSN3252
   27 CONTINUE                                                          ANSN3253
   26 CONTINUE                                                          ANSN3254
```

Figure B1 (contd.)

```
   34 CONTINUE                                                          ANSN3255
      DO 29 J=1,IZP                                                     ANSN3256
      CT(J)=0.0
      DO 29 IIG=1,IGP                                                   ANSN3257
      E1=FXS(IIG,J) + FIS(IIG,J) + SNN(IIG,J) + AMAX1(XLL(IIG,J),0.0)
     *  - AMIN1(RLK(IIG,J),0.0)
      E2=OTS(IIG,J) + APS(IIG,J) + AMAX1(RLK(IIG,J),0.0) -
     *     AMIN1(XLL(IIG,J),0.0)
   29 XEB(IIG,J)=E1/E2                                                  ANSN3270
      DO 1029 I=1,IM
      J=MA(I)
      CT(J)=CT(J) + V(I)
 1029 CT(IZP)=CT(IZP) + V(I)
      DO 38 J=1,IZP                                                     ANSN3271
   32 IF(J.EQ.IZP)GO TO 31                                              ANSN3272
C     *** PRINT SUMMARY TABLES                                          ANSN3273
      WRITE (NOU,20) J,IGP,CT(J)
   20 FORMAT(18H1 SUMMARY FOR ZONE I3,47H BY GROUP INCLUDING SUM FOR ALLANSN3275
     1 GROUPS IN LINE I3,'    VOLUME=',1PE13.5)
      GO TO 33                                                          ANSN3277
   31 WRITE (NOU,30) CT(IZP)
   30 FORMAT(20H1 SUMMARY FOR SYSTEM,'    VOLUME=',1PE13.5)
   33 WRITE (NOU,40) (I,FXS(I,J),FIS(I,J),SNN(I,J),SFS(I,J),OTS(I,J),APSANSN3280
     1(I,J),XLK(I,J),XBB(I,J),I=1,IGP)                                  ANSN3281
   40 FORMAT(108HO GRP.  FIX  SOURCE  FISS SOURCE  IN  SCATTER  SLF SCATANSN3282
     1TER  OUT SCATTER   ABSORPTION    LEAKAGE      BALANCE/(I6,1P8E13.5ANSN3283
     2))                                                                ANSN3284
      WRITE (NOU,50) (I,RXN(I,J),RFL(I,J),RCT(I,J),RLK(I,J),XLL(I,J),FRTANSN3285
     1(I,J),TXN(I,J),DEN(I,J),I=1,IGP)                                  ANSN3286
   50 FORMAT(108HO GRP.  RT BDY FLUX   RT BDY J+    RT BDY J     RT LEAKANSN3287
     1AGE  LFT LEAKAGE   FISS RATE    TOTAL FLUX    DENSITY/(I6,1P8E13.5ANSN3288
     2))                                                                ANSN3289
   38 CONTINUE                                                          ANSN3290
      IF(ID.LT.1000000)GO TO 999                                        ANSN3291
      CALL FACTOR(MZ,D(LMB),D(LMC),D(LXMD),MA,V,XN,DF,TAB,IM,IGM)       ANSN3292
  999 RETURN                                                            ANSN3293
      END                                                               ANSN3294
//LKED.PROG DD DISP=SHR,DSN=X.WAR14636.WARPROG
 INCLUDE PROG(ANISN)
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
//   DCB=(RECFM=VBS,LRECL=X,BLKSIZE=7208)
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
//   DCB=(RECFM=VBS,LRECL=X,BLKSIZE=7208)
//GO.FT03F001 DD UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
//   DCB=(RECFM=VBS,LRECL=X,BLKSIZE=7208)
//GO.FT04F001 DD DSN=T.TJH45255.XSPU,DISP=SHR
//GO.FT07F001 DD SYSOUT=B
//GO.FT08F001 DD UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
//   DCB=(RECFM=VBS,LRECL=X,BLKSIZE=7208)
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(TRK,(100,10),RLSE),
//   DCB=(RECFM=VBS,LRECL=X,BLKSIZE=7208)
//GO.FT05F001 DD *
```

Figure B1 (contd.)

```
SPUTTERING:  H->NI   20GPS
15$$ 1 0 19 32 1 0 0 1 200 0 20 3 4 23 20 0 20 20 0 0 0 1 200 30 0 0
 0 0 1 0 0 0 3 1 0 0
16** 2R0.0 0.001 4R0.0 1.0 0.0 0.5 0.0001 3R0.0  T
13$$ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
  T
18** 0.0 1.0 F0.0  T
3**  F0.0  T
1**  F0.0
4** 99I0.0 99I1000. 2000.
5**  F1.0
6**
 0. .0067881 .0155634 .0237896 .0311572 .037399 .0422691 .0456506
 .0473626 .0473626 .0456508 .0422891 .037399 .0311572
 .0237896 .0155634 .0067881 1N16
7**
 -1. -.9947 -.972286 -.932815 -.877701 -.808937 -.729008 -.640801
 -.547506 -.452494 -.359198 -.270992 -.191062 -.122298 -.0671843
 -.0277125 -.00529954 1N16
8$$ F1
9$$ 1
10$$ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
11$$ F0
12** F1.0
19$$ 19
  T  T

/*
//
```

Figure B1 (contd.)