

AD-A257 986



RL-TR-92-48

Final Technical Report

April 1992



METHODS DEVELOPMENT FOR ELECTRON TRANSPORT

University of Arizona

Barry D. Ganapol

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

92-30963



398603

1458Y

**DTIC
SELECTE
DEC 08 1992
S B D**

**Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700**

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-92-48 has been reviewed and is approved for publication.

APPROVED:



JOHN C. GARTH
Project Engineer

FOR THE COMMANDER:



HAROLD ROTH, Director
Solid State Sciences

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL(ERT) Hanscom AFB MA 01731-5000. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE April 1992		3. REPORT TYPE AND DATES COVERED Final May 1990 - May 1991	
4. TITLE AND SUBTITLE METHODS DEVELOPMENT FOR ELECTRON TRANSPORT				5. FUNDING NUMBERS C (15) F30602-88-D-0026 Task S-0-7543 PE - 61102F PR - 2306 TA - J3 WC - P8	
6. AUTHOR(S) Barry D. Ganapol				8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Arizona Department of Nuclear & Energy Engineering Tucson AZ 85721				10. SPONSORING/MONITORING AGENCY REPORT NUMBER RL-TR-92-48	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Laboratory (ERT) Hanscom AFB MA 01731-5000					
11. SUPPLEMENTARY NOTES Rome Laboratory Project Engineer: John C. Garth/ERT/(617) 377-2360. The Prime Contractor is Calspan/UB Research Center, PO Box 400, Buffalo NY 14225.					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report consists of two code manuals and an article recently published in the proceedings of the American Nuclear Society Mathematics and Computation Topical Meeting held in Pittsburgh. In these presentations, deterministic calculational methods simulating electron transport in solids are detailed. The first method presented (Section 2) is for the solution of the Spencer-Lewis equation in which Electron Motion is characterized by continuous slowing down theory and a pathlength formulation. The F_N solution to the standard monoenergetic transport equation for electron transport with isotropic scattering in finite media is given in Section 3. For both codes, complete flow charts, operational instructions and sample problems are included. Finally, in Section 4, an application of the multigroup formulation of electron transport in an infinite medium is used to verify an equivalent S_N formulation. For this case, anisotropic scattering is also included.					
14. SUBJECT TERMS Electron Transport, Computer Codes, Boltzmann Transport Equation (Analytic Solution), The F_N Method				15. NUMBER OF PAGES 142	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL		

1 Introduction

The final report to follow is a compilation of two reports and an article recently published in the proceedings of the 1991 American Nuclear Society (ANS) Topical Meeting of the Mathematics and Computation (M&C) Division. The three publications deal with the development of deterministic calculational methods for three different physical characterizations of the motion of electrons in a solid as given by the Boltzmann transport equation. These methods are the initial step in developing robust algorithms required to eventually predict the response of integrated circuits to incident electron and gamma showers. Highly accurate numerical solutions have been generated finding use as benchmarks to which algorithms, developed for realistic applications, are to be compared. The reports not only include the mathematical theory associated with the solution to the appropriate transport equation, but also the computational strategy and operational instructions for the accompanying computer code. A computer code is briefly described in the article for the ANS Topical Meeting with a full description to be published elsewhere.

In characterizing the motion of electrons in matter, one is faced with a spectrum of physical phenomena all requiring a model. For instance, in the plasma physics application, collective or long range forces play a dominate role. For this reason, a Vlasov approximation, where collisions play only a minor role, is the appropriate form of the transport equation. In the solid state regime, where collisions provide the forces for electron motion at high energy and the forces from fields are relatively weak, a linear transport equation provides an adequate description. In this formulation, the major assumption is that the atomic electric fields are localized (within the Debye sphere) allowing the application of the collision/streaming statistical model given by the linear Boltzmann equation. There exist further approximations which lead to several simplified, but useful, models such as:

- the Spencer-Lewis (S-L) equations,
- the monoenergetic transport equation, and
- the multigroup transport equation.

These are the three specific areas detailed in the following chapters in this report.

In the S-L formulation (in §2), the continuous slowing down approximation (CSDA) is enforced where electrons are restricted to a specific energy loss for a given penetration distance or pathlength. When the interaction probabilities are assumed constant, in addition, the solution to the S-L equation can be obtained from a combination of a multiple collision

expansion and a moments reconstruction. The analytical representation in terms of path-length can be evaluated to an exceptionally high accuracy to establish a true benchmark.

The monoenergetic transport equation in a finite medium (in §3) is solved with the highly accurate F_N method. The algorithm features heterogeneous media with a uniform isotropically emitting source in each slab. Currently, this algorithm is limited to isotropic scattering with the intent to include anisotropic scattering and down scattering in the more advance multigroup version.

The final presentation (in §4) concerns the multigroup formulation of electron down scattering in an infinite medium. The solution with anisotropic scattering is obtained via a numerical Fourier transform. This accurate multigroup benchmark is compared to a standard discrete ordinates approximation from which the order of the error for various spatial approximations relating the average to the cell-edge flux is confirmed.

2 SLEET - The Spencer-Lewis Equation of Electron Transport: A Code Manual

Abstract

I. Introduction

II. The Transport Description

- A. The Spencer-Lewis (S-L) Equation
- B. Extended Transport Correction
- C. Screened-Rutherford Scattering Kernel
- D. Source Distribution

III. Reconstruction of the Scalar Density

- A. Multiple Collision Formulation
- B. Legendre Series Expansion for $\psi_n(\eta)$
- C. Spatially Distributed Source
- D. Power Law Variation of λ^{-1}
- E. Uncollided and First Collided Fluxes

IV. Numerical Evaluation

- A. Expansion Coefficients
- B. Convergence Iteration
- C. Convergence Acceleration

V. The SLEET Code

- A. Input Options
- B. Sample Problems

Problem 1

Problem 2

Problem 3

- C. Programming Notes

References

Appendix

The SLEET Code

DTIC QUALITY INSPECTED 2

A1

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

SLEET

The Spencer-Lewis Equation of Electron Transport: A Code Manual

**B. Ganapol†
Department of Nuclear and
Energy Engineering**

**University of Arizona
Tucson, Arizona**

Abstract

The Spencer-Lewis equation of electron transport with energy independent cross sections and a screened-Rutherford scattering kernel in an infinite medium is solved using moments reconstruction coupled with the multiple collision formulation. The solution, evaluated to a high degree of accuracy, finds use as a benchmark against which purely numerically based algorithms can be tested. The evaluation features general anisotropic scattering for a localized source or for a uniformly distributed source in a half-space. The operation of the SLEET code which embodies the numerical solution is described in detail in this document.

**†Work performed for Rome Air Development Center at Hanscom AFB
CALSPAN.
Contract No. F30602-88-D-0026, S-0-7542**

I. INTRODUCTION

With the ever decreasing dimensions of integrated circuits and the necessity for shielding these sensitive electronic components against penetrating electrons, a renewed interest in electron transport theory has recently been inspired. Because of the complex nature of electron transport in solids, direct numerical simulation of electron motion by Monte Carlo calculations has found widespread use. Monte Carlo simulations are flexible and can incorporate a variety of physical processes in complicated three-dimensional configurations. At the same time however, probabilistic analyses can require excessive computational effort and therefore, may not be the most efficient calculational tool for all problems. Thus, new approaches leading to a numerical characterization of electron transport in solids have and will continue to be developed.

In a classic paper, Lewis¹ proposed a Boltzmann equation description of electron motion in terms of path length. In this formulation, angular deflection is specified by a scattering term representing deflection without energy loss; and energy degradation is characterized by a prescribed range-energy relation. Lewis's description assumed that the electrons continuously lose energy with motion thus the statistical nature of the slowing down process is neglected. The resulting Boltzmann equation, called the Spencer-Lewis equation, was first solved numerically by Spencer² for an infinite homogeneous medium by using moments to reconstruct the electron dose from a functional fit based on physical arguments. In addition to the moment function fitting procedure of Spencer, a significant effort has been given to the application of S_N methods to these problems.³

The development of S_N and other numerical methods for use in electron transport applications has given rise for the need to assess the accuracy of proposed numerical algorithms and to provide the assurance that these algorithms have been properly programmed. Analytical solutions for representative electron transport problems which can be evaluated to a high degree of accuracy serve as standards or benchmarks for comparison with the approximate solutions. Such a solution has been developed for the S-L equation characterizing electron transport in a simplified physical setting. Specifically, electrons are assumed to slow down in an infinite medium after emission. Furthermore, all scattering and absorption interactions are assumed to be energy independent (except when otherwise stated) with the kinematics of the scattering

process described by a screened-Rutherford law.

The method of numerical solution employs a flux reconstruction technique a la Spencer, however, the functional form of the desired flux distribution is not known a priori. The reconstruction is based on an expansion in Legendre polynomials coupled with the multiple collision formulation. In this way, the solution can be evaluated to a high degree of accuracy (4-5 digits). The SLEET (Spencer-Lewis Equation of Electron Transport) program has been written specifically for this purpose. This report is a manual describing the SLEET code in detail including theory and operation.

II. THE TRANSPORT DESCRIPTION

A. The Spencer-Lewis (S-L) Equation

The S-L equation describing the motion of electrons experiencing deflection and energy loss in one-dimensional plane homogeneous medium with azimuthal symmetry is

$$\left[\frac{\partial}{\partial s} + \mu \frac{\partial}{\partial x} + \frac{1}{\lambda(s)} \right] f(x, \mu, s) = \frac{1}{\lambda(s)} \int_0^{2\pi} d\phi' \int_{-1}^1 d\mu' p(\mu_0, s) f(x, \mu', s) + q(x, \mu, s) \quad (1)$$

where

s = cumulative electron path length

x = distance measured along coordinate axis

f = angular density of electrons

λ = total mean free path

$p(\mu_0, s)d\mu_0$ = probability of an elastic scattering event producing a deflection between μ_0 and $\mu_0 + d\mu_0$

ϕ = azimuthal angle

$q \, dx \, d\mu \, ds$ = volumetric rate of external injection of electrons at position x in direction μ with path length s .

In the customary fashion, the scattering kernel is assumed to be expandable in terms of Legendre polynomials P_L :

$$p(\mu_0, s) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \omega_{\ell}(s) P_{\ell}(\mu_0) , \omega_0 = 1 \quad (2)$$

and Eq. (1) becomes

$$\left[\frac{\partial}{\partial s} + \mu \frac{\partial}{\partial x} + \frac{1}{\lambda(s)} \right] f(x, \mu, s) = \frac{1}{\lambda(s)} \int_{-1}^1 d\mu' g(\mu' \rightarrow \mu, s) f(x, \mu', s) + q(x, \mu, s) \quad (3a)$$

where

$$g(\mu' \rightarrow \mu, s) \equiv \sum_{\ell=0}^{\infty} \frac{2\ell+1}{2} \omega_{\ell}(s) P_{\ell}(\mu) P_{\ell}(\mu') . \quad (3b)$$

This equation will be solved in an infinite medium with constant properties and with the following boundary condition (for a localized source);

$$\lim_{|x| \rightarrow \infty} f(x, \mu, s) = 0 \quad (3c)$$

In addition, electrons are physically restricted to positive path lengths requiring

$$f(x, \mu, s) = 0, s < 0 . \quad (3d)$$

B. Extended Transport Correction

Typically in electron transport applications, the series representation for the scattering kernel given by Eq. (3b) requires many terms to provide an adequate description of the scattering event. An extended transport correction⁴ has been proposed as a way to reduce the number of terms in the scattering approximation while maintaining the desired accuracy. In particular, the series for $g(\mu' \rightarrow \mu)$ is replaced by a series with modified coefficients ω_{ℓ}^* truncated at L plus a delta function to represent the forward directed component; thus Eq. (3a) becomes

$$\left[\frac{\partial}{\partial s} + \mu \frac{\partial}{\partial x} + \frac{1}{\lambda^*} \right] f(x, \mu, s) = \frac{1}{\lambda^*} \int_{-1}^1 d\mu' g^*(\mu' \rightarrow \mu) f(x, \mu', s) + q(x, \mu, s) \quad (4a)$$

with the extended transport corrected kernel and mean free path given by

$$g^*(\mu' \rightarrow \mu) \equiv \sum_{\ell=0}^L \frac{2\ell+1}{2} \omega_{\ell}^* P_{\ell}(\mu) P_{\ell}(\mu') \quad (4b)$$

$$\omega_{\ell}^* = [\omega_{\ell} - \omega_{L+1}] / [1 - \omega_{L+1}] , 0 \leq \ell \leq L \quad (4c)$$

$$\left(\frac{1}{\lambda^*}\right)^{-1} = \left(\frac{1-\omega_{\ell+1}}{\lambda}\right)^{-1} \quad (4d)$$

C. Screened-Rutherford Scattering Kernel

Since electrons are negatively charged particles, they are deflected by the electric fields associated with the electronic cloud as well as the nucleus of the host medium atoms. For this situation, the screened-Rutherford scattering kernel is an appropriate approximation with the following Legendre coefficients derived in Appendix A:

$$\omega_0 = 1 \quad (5a)$$

$$\omega_1 = 1 + \hat{\eta} - \hat{\eta}(1+\hat{\eta}/2) \ln(1+2/\hat{\eta}) \quad (5b)$$

$$\ell\omega_{\ell+1} = (2\ell+1)(1+\hat{\eta})\omega_{\ell} - (\ell+1)\omega_{\ell-1} \quad (5c)$$

and

$$\hat{\eta} = 3.2 \times 10^{-3} z^{2/3} E(\text{keV}) \quad (5d)$$

$$\frac{1}{\lambda} = 2\pi \frac{Z(Z+1)}{A} \frac{N_A r_e^2}{4} \frac{E_e}{\bar{E}(\text{MeV})^2} \frac{1}{1+\hat{\eta}/2}$$

where

E_e = electron rest mass energy (0.511 MeV)

$\bar{E}$ = average electron energy

N_A = Avogadro's number

A = atomic weight of the scattering center

Z = atomic number of scattering center

r_e = electron radius (2.8179×10^{-13} cm).

D. Source Distribution

The source is assumed to be located at the zero plane ($x=0$) and to emit electrons having a zero path length with an angular distribution specified by $Q(\mu)$, thus

$$q(x, \mu, s) = Q(\mu) \delta(x) \delta(s).$$

The angular source distribution will be of two types

$$Q(\mu) = \begin{cases} 1/2, & \text{isotropic emission} \\ Q(\mu-\mu_0), & \text{azimuthally symmetric beam emission} \end{cases} \quad (6a)$$

(6b)

where μ_0 is the cosine of the angle of incidence of the source electrons.

III. RECONSTRUCTION OF THE SCALAR DENSITY

A. Multiple Collision Formulation

By decomposition of $f(x, \mu, s)$ into its collisional components f_n , Eqs. (4a), (3c) and (3d) are transformed into the following multiple collision representation:

$$\mathbb{L} f_0(x, \mu, s) = Q(\mu) \delta(x) \delta(s) \quad (7a)$$

$$\mathbb{L} f_n(x, \mu, s) = \int_{-1}^1 d\mu' g^*(\mu' \rightarrow \mu) f_{n-1}(x, \mu', s) \quad , n = 1, 2, \dots \quad (7b)$$

$$\lim_{|x| \rightarrow \infty} f_n(x, \mu, s) = 0 \quad (7c)$$

$$\mathbb{L} \equiv \frac{\partial}{\partial s} + \mu \frac{\partial}{\partial x} + \frac{1}{\lambda^*}$$

with the total angular flux given by

$$f(x, \mu, s) = \sum_{n=0}^{\infty} f_n(x, \mu, s) . \quad (7d)$$

By application of integral transport theory and the definition of the similarity variable $\eta \equiv x/s$, the scalar flux solutions to Eqs. (7) for $n = 0, 1$ are

$$f_0(x, s) = \frac{e^{-s/\lambda^*}}{s} \psi_0(\eta) \theta(1-|\eta|) \quad (8a)$$

where

$$\psi_0(\eta) \equiv Q(\eta) \quad , \quad (8b)$$

and

$$f_1(x,s) = e^{-s/\lambda^*} \psi_1(\eta) \theta(1-|\eta|) \quad (8c)$$

with

$$\psi_1(\eta) \equiv \int_{-1}^{\eta} d\mu \int_{\eta}^1 d\eta' \frac{Q(\eta')}{\eta' - \mu} g^*(\eta' \rightarrow \mu) + \int_{\eta}^1 d\mu \int_{-1}^{\eta} d\eta' \frac{Q(\eta')}{\mu - \eta'} g^*(\eta' \rightarrow \mu) \quad (8d)$$

$\theta(u)$ is the Heaviside step function. From the correspondence with the time-dependent neutron transport case⁵, it can be shown that

$$f_n(x,\mu,s) = \frac{e^{-s/\lambda^*}}{s} \frac{(s/\lambda^*)^n}{n!} F_n(\mu,\eta) \theta(1-|\eta|) \quad (9)$$

where F_n satisfies the reduced collision equation

$$\left[(\mu - \eta) \frac{\partial}{\partial \eta} + n - 1 \right] F_n(\mu,\eta) = n \int_{-1}^1 d\mu' g^*(\mu' \rightarrow \mu) F_{n-1}(\mu',\eta) \quad (10a)$$

for $n \geq 1$ with the condition

$$\lim_{|\eta| \rightarrow 1} F_n(\mu,\eta) = 0 \quad (10b)$$

Thus the scalar density to be evaluated is given by

$$f(x,s) = \frac{e^{-s/\lambda^*}}{s} \sum_{n=0}^{\infty} \frac{(s/\lambda^*)^n}{n!} \psi_n(\eta) \theta(1-|\eta|) \quad (11a)$$

with

$$\psi_n(\eta) \equiv \int_{-1}^1 d\mu' F_n(\mu',\eta) \quad (11b)$$

B. Legendre Series Expansion for $\psi_n(\eta)$

Since the collisional process, even for highly forward peaked scattering, provides a mechanism for redistribution of particles in phase space and therefore a smoothing of highly directed sources, F_n as a function of μ and η (for n sufficiently large) is expected to be continuous and uniformly bounded. By taking this behavior into account and since F_n is nonzero for $|\eta| \leq 1$, the following expansion of F_n in Legendre polynomials in the variable η seems entirely appropriate:

$$F_n(\mu, \eta) = \sum_{k=0}^{\infty} \frac{2k+1}{2} f_{n,k}(\mu) P_k(\eta) \quad (12a)$$

where

$$f_{n,k}(\mu) \equiv \int_{-1}^1 d\eta P_k(\eta) F_n(\mu, \eta); \quad (12b)$$

and for the scalar component

$$\psi_n(\eta) = \sum_{k=0}^{\infty} \frac{2k+1}{2} f_{n,k}^0 P_k(\eta), \quad (13a)$$

where the coefficients $f_{n,k}^0$ are

$$f_{n,k}^0 = \int_{-1}^1 d\mu f_{n,k}(\eta) \psi_n(\eta). \quad (13b)$$

The expansion coefficients are found directly from Eq. (10a) by substitution of Eq. (12a), multiplying by $P_k(\eta) P_\ell(\mu)$ and integrating over $\eta [-1, 1]$ and $\mu [-1, 1]$ to yield the recursion relation

$$f_{n,0}^\ell = (\omega_\ell^n) \int_{-1}^1 d\eta Q(\eta) P_\ell(\eta) \quad (14a)$$

$$f_{0,k}^\ell = \int_{-1}^1 d\eta P_\ell(\eta) P_k(\eta) Q(\eta) \quad (14b)$$

and for $k \geq 1, n \geq 1$

$$\begin{aligned} (k+n) f_{n,k}^\ell &= - (k-n-1) f_{n,k-2}^\ell + \frac{2k-1}{2\ell+1} [\ell f_{n,k-1}^{\ell-1} + (\ell+1) f_{n,k-1}^{\ell+1}] + \\ &+ n \omega_\ell^\ell [f_{n-1,k}^\ell - f_{n-1,k-2}^\ell] \end{aligned} \quad (14c)$$

with

$$f_{n,-1}^\ell \equiv 0. \quad (14d)$$

C. Spatially Distributed Source

The scalar density $h(x,s)$ for a spatially distributed source of the form

$$q(x, \mu, s) = U(x) Q(\mu) \delta(s) \quad (15a)$$

can be represented by the convolution integral

$$h(x,s) = \int_{-\infty}^{\infty} dx' U(x-x') f(x',s) \quad (15b)$$

where $f(x,s)$ is given by Eq. (11a). Thus Eq. (15b) becomes, after substitution of Eqs. (11a) and (13a),

$$h(x,s) = \frac{1}{\lambda^*} \sum_{k=0}^{\infty} \frac{2k+1}{2} e^{-s/\lambda^*} \sum_{n=0}^{\infty} (s/\lambda^*)^n f_{n,k}^0 I_k(\eta) \quad (16a)$$

where

$$I_k(\eta) = \int_{-1}^1 d\eta' U(s-x\eta') P_k(\eta') \quad (16b)$$

D. Power Law Variation of λ^{*-1}

If λ^{*-1} is assumed to be of the form

$$\lambda^{*-1}(s) = \lambda_0(s/s_0)^\beta, \quad -1 < \beta < \infty, \quad (17)$$

then the solutions obtained thus far can be shown to have the same representation with the following replacement

$$e^{s/\lambda^*} \rightarrow \exp \left[- \frac{s}{1+\beta} \left(\frac{s}{s_0} \right)^\beta \right]$$

$$f_{n,k}^0 \rightarrow \left(\frac{s}{s_0} \right)^{n\beta} f_{n,k}^0(\beta)$$

and

$$f_{n,0}^\ell(\beta) = \left(\frac{\omega_\ell}{1+\beta} \right)^n f_{0,0}^\ell \quad (18a)$$

$$[k+n(1+\beta)] f_{n,k}^\ell(\beta) = -[k-n(1+\beta)-1] f_{n,k-1}^\ell(\beta) + \frac{2k-1}{2\ell+1} [\ell f_{n,k-1}^{\ell-1}(\beta) + (\ell+1) f_{n,k-1}^{\ell+1}(\beta)] + n \omega_\ell [f_{n-1,k}^\ell(\beta) - f_{n-1,k-2}^\ell(\beta)] \quad (18b)$$

$$f_{0,k}^\ell(\beta) = f_{0,k}^\ell \quad (18c)$$

E. Uncollided and First Collided Fluxes

As a result of the localized nature of the source in x and s , the uncollided scalar flux and possibly the first collided flux are singular. Since the Legendre expansion converges slowly near singularities, it is advantageous to remove these components from the expansion. This is done by obtaining the uncollided and first collided fluxes analytically from the solution of Eqs. (7a) and (7b) for $n = 1$ (for a beam source) to give for an isotropic source [$Q(\mu) = Q_0/2$]

$$f_0(x,s) = \frac{Q_0}{2} \frac{e^{-s/\lambda^*}}{2s} \theta(1-|\eta|) \quad (19a)$$

and for a beam source

$$f_0(x,s) = Q_0 \frac{e^{-s/\lambda^*}}{s} \delta(\eta - \mu_0) \quad (19b)$$

$$f_1(x,s) = Q_0 \frac{e^{-s/\lambda^*}}{\lambda^*} [\theta(\mu_0 - \eta) J^-(\eta) + \theta(\eta - \mu_0) J^+(\eta)] \quad (19c)$$

$$J^\pm(\eta) \equiv \sum_{\ell=0}^L \frac{2\ell+1}{2} \omega_\ell P_\ell(\mu_0) Y_\ell^\pm \quad (19d)$$

$$Y_0^\pm = \ln(1 \pm \mu) / (\mu_0 - \eta) \quad (19e)$$

$$Y_1^\pm = \pm(1 \mp \eta) + \mu_0 Y_0^\pm \quad (19f)$$

$$Y_\ell^\pm = \frac{2\ell-1}{\ell} \mu_0 Y_{\ell-1}^\pm - \frac{\ell-1}{\ell} Y_{\ell-2}^\pm - \frac{1}{\ell} [P_\ell(\eta) - P_{\ell-2}(\eta)] \quad (19g)$$

For a uniformly distributed isotropic source for $x < 0$

$$f_0(x,s) = Q_0 \cdot \begin{cases} e^{-s/\lambda^*} & , \eta < -1 \\ e^{-s/\lambda^*} (1-\eta)/2 & , |\eta| \leq 1 \\ 0 & , \eta > 1 \end{cases} \quad (20a)$$

and for a beam source

$$f_0(x,s) = Q_0 \cdot \begin{cases} e^{-s/\lambda^*} & , \eta < -1 \\ e^{-s/\lambda^*} \sigma(\mu_0 - \eta) & , |\eta| \leq 1 \\ 0 & , \eta > 1 \end{cases} \quad (20b)$$

IV. Numerical Evaluation

The expression for the scalar density obtained for a localized source at $x = 0$ is

$$f(x,s) = \sum_{n=0}^{n_s-1} f_n(x,s) + \sum_{k=0}^{\infty} \frac{2k+1}{2} f_k(s) P_k(\eta) \quad (21a)$$

where the discontinuous uncollided ($n = 0$) and first collided ($n = 1$) densities have been removed from the Legendre expansion in order to expedite convergence and

$$f_k(s) = \frac{e^{-s/\lambda^*}}{s} \sum_{n=n_s}^{\infty} \frac{(s/\lambda^*)^n}{n!} f_{n,k}^0 \quad (21b)$$

A further simplification can be realized by noting that for $n_s = 2$

$$f_0(s) = \frac{1}{s} M_0(s) - \left[f_{0,0}^0 + \frac{s}{\lambda^*} f_{1,0}^0 \right] \frac{e^{-s/\lambda^*}}{s} \quad (22)$$

where

$$M_0(s) = \int_{-\infty}^{\infty} dx f(x,s)$$

From particle conservation $M_0(s) = 1$, and from Eq. (14a)

$$f_{n,0}^0 = 1$$

thus

$$f_0(s) = \left[1 - \left(1 + \frac{s}{\lambda^*} \right) e^{-s/\lambda^*} \right] / s \quad (23)$$

for a beam source and

$$f_0(s) = (1 - e^{-s/\lambda^*}) / s \quad (23b)$$

for an isotropic source. The final expression to be evaluated is therefore

$$f(x,s) = \sum_{n=0}^{n_s-1} f_n(s) + \sum_{k=1}^{\infty} \frac{2k+1}{2} f_k(s) P_k(\eta) \quad (24)$$

To evaluate this expression the following numerical operations are required:

- recursive evaluation of $f_{n,k}^0$
- multiple collision series summation (sum over n)
- moments reconstruction (sum over k) .

A. Expansion Coefficients

The major numerical error associated with the recursion relation for $f_{n,k}^0$ is round-off. To determine $f_{n,k}^0$, the coefficients for

$$n = 0, 1, \dots, L_N$$

$$k = 0, 1, \dots, L_K$$

$$\ell = 0, 1, \dots, L_K$$

are required due to the telescoping nature of the recurrence relation. L_N and L_K are the maximum number of terms allowed in the multiple collision and Legendre series evaluation respectively.

B. Convergence Iteration

The infinite series are evaluated using a simple engineering estimate where the series is considered converged when each of IEK terms produces a relative error of less than a specified amount (ERN and ERK for the multiple collision and Legendre series respectively). The multiple collision series is evaluated first (for each x and s) for a truncation error $ERN = ERK/10$. If the series converges, a further test is performed to determine whether the multiple collision summation (over n) has been evaluated with sufficient digits to give the desired accuracy. This is accomplished by comparison of each term of the Legendre sum with the final result. If the final outcome has been obtained as a result of addition and subtraction of much larger numbers than the final result, the flux may largely be noise if the individual terms did not contain enough digits. If true, a second iteration begins where the multiple collision sum is recalculated with a reduced truncation error $ERN/10^r$, where r is the estimated number of digits required, and the Legendre series is again summed. The process is continued until $ERN < 10^{-20}$ or the flux changes less than ERK between successive iterations.

C. Convergence Acceleration

For highly anisotropic scattering, the Legendre series can be slowly convergent or even numerically divergent. The divergence results from round-off error that produces inaccuracies in the terms of the summation eventually leading to numerical noise and subsequent divergence. For isotropic and beam

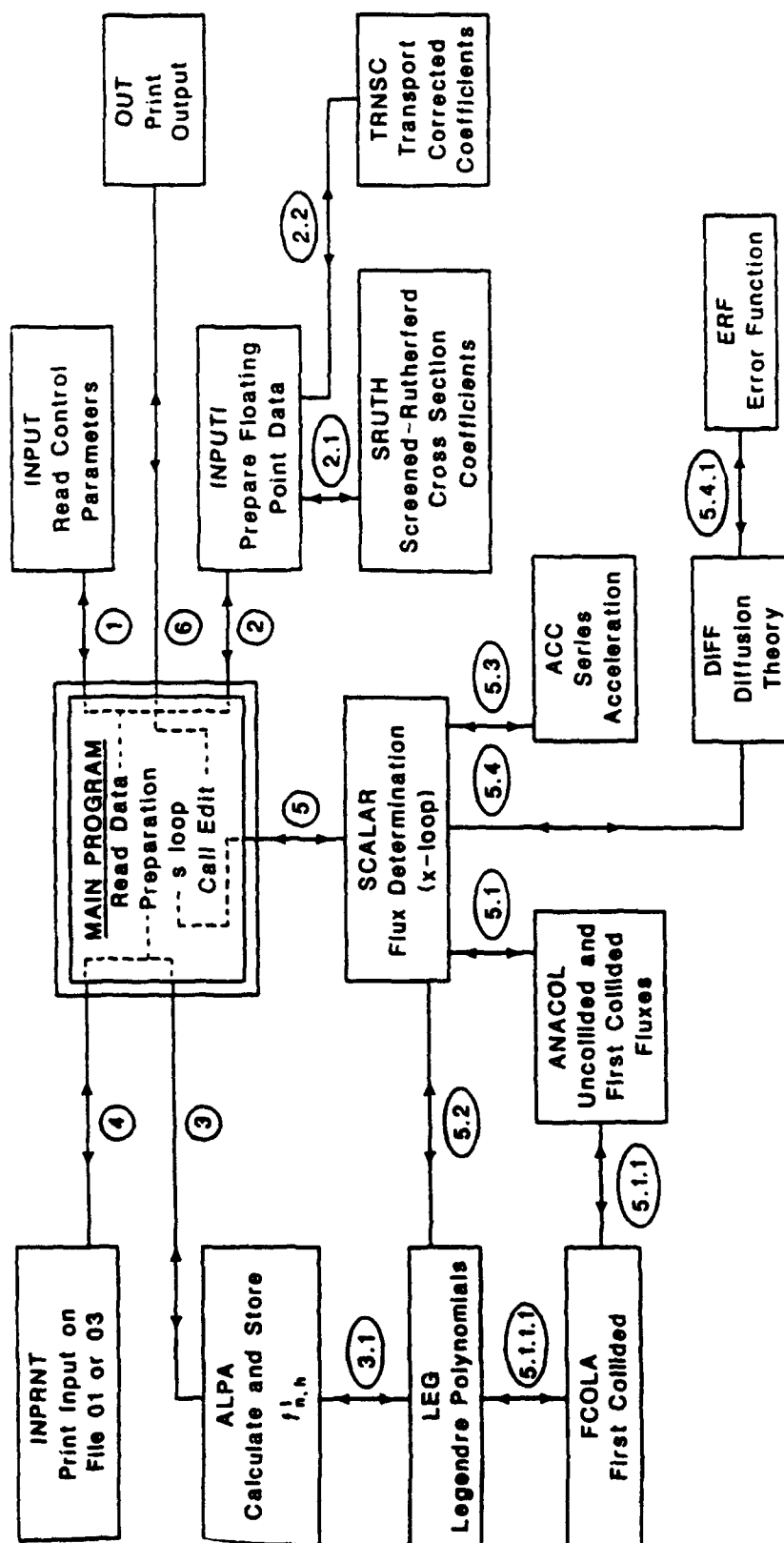


Fig. 1 SLEET logic flow chart

sources (at $x=0$), the convergence of the Legendre series can be accelerated by removal of the singularities at $\eta = \pm 1$. This is accomplished by representation of the flux that has had at least one collision by

$$(1-\eta^2)^m [f(x,s) - f_0(x,s)] = \sum_{k=0}^{\infty} \frac{2k+1}{2} A_k^{(m)}(k) P_k(\eta) \quad (25a)$$

The motivation for this expansion is that the n th derivative of $\psi_n^{(m)}$ possesses singularities at $\eta = \pm 1$ and the above expansion tends to remove these singularities. The expansion coefficients are obtained from the recursion relation:

$$\begin{aligned} A_k^{(0)}(s) &= f_k(s) \\ A_k^{(m)}(s) &= (1-\gamma_{2k}) A_k^{(m-1)} - \gamma_{1k} A_{k+2}^{(m-1)} - \gamma_{3k} A_{k-1}^{(m-1)} \\ \gamma_{1k} &\equiv (k+1)(k+2) / (2k+1)(2k+3) \\ \gamma_{2k} &\equiv \frac{(k+1)^2}{(2k+1)(2k+3)} + \frac{k^2}{(2k-1)(2k+1)} \\ \gamma_{3k} &\equiv (k+1)k / (2k-1)(2k+1) \end{aligned} \quad (25b)$$

Because the function expanded is smoother than $f(x,s)$ itself, the series in Eq. (22) will, in general, converge more rapidly than the series in Eq. (21a).

V. The SLEET Code

The algorithm described above was coded in a FORTRAN 77 program called SLEET which is an acronym for the Spencer-Lewis Equation of Electron Transport. A flow diagram indicating the logic of the program is given in Fig. 1. Each cell has been given a number indicating the calling sequence allowing the original call to be traced. The program is composed of 15 routines each with a specific function summarized in the flow chart.

A. Input Options

The SLEET input description is given in Fig. 2. the input file is called SLEET.DAT and must exist at execution. Free format is used where the entries on each line must be separated by at least a space.

The following options are available:

- 1) Isotropic or beam source angular distributions.
- 2) Delta function or uniform half-space ($x \leq 0$) spatial source distributions

Fig. 2. SLEET input

```

C*****
C INPUT DESCRIPTION
C
C CARD 1 (18A4)
C IDEN = PROBLEM IDENTIFICATION
C
C CARD 2
C IS = 1 ISOTROPIC SOURCE
C      = 2 ANISOTROPIC SOURCE
C
C JTI = 0 DELTA FUNCTION SOURCE AT
C       X=0
C      = 1 CONSTANT SOURCE IN LEFT
C          HALF-SPACE(X<0)
C
C Q0 = SOURCE STRENGTH
C
C AM0 = COSINE OF ANGLE OF INCIDENCE OF SOURCE
C       ELECTRONS
C
C CARD 3
C LL = ORDER OF SCATTERING KERNEL APPROXIMATION
C
C ITC --1 SCREENED-RUTHERFORD SCATTERING KERNEL
C      = 1 READ IN SCATTERING COEFFICIENTS
C
C ITRN --1 USE TRANSPORT CORRECTED CROSS SECTIONS
C        = 1 DO NOT USE TRANSPORT CORRECTED CROSS
C          SECTIONS
C
C CARD 4A IF ITC=1
C ALA = TOTAL MEAN FREE PATH
C
C CARD 4B IF ITC=-1
C ETA = SCREENING PARAMETER FOR SCREENED
C       RUTHERFORD SCATTERING CROSS SECTION
C
C ALA = TOTAL MEAN FREE PATH
C
C CARD 5
C BE = POWER OF PATH DEPENDENCE OF NUMBER
C      OF SECONDARIES (ALA(S)=(S/SI0)**BE)
C SI0 = CROSS SECTION PARAMETER
C
C CARD 6
C LS = NUMBER OF PATH GRID POINTS
C
C LX = NUMBER OF SPACE GRID POINTS
C
C JT --1 READ IN PATH GRID POINTS
C      = 1 READ IN PATH MESH SPACING
C
C JX --1 READ IN SPACE GRID POINTS
C      = 1 READ IN SPACE MESH SPACING
C

```

C CARD 7
 C IDFF--1 COMPARISON WITH DIFFUSION THEORY
 C (BE=0 ONLY)
 C = 1 SCALAR FLUX CALCULATION
 C
 C CARD 8
 C LN = MAXIMUM NUMBER OF TERMS IN
 C MULTIPLE COLLISION EXPANSION
 C OF MOMENTS
 C
 C LK = MAXIMUM NUMBER OF TERMS IN
 C LEGENDRE SERIES EXPANSION
 C
 C IEK = NUMBER OF CONSECUTIVE TERMS
 C SATISFYING RELATIVE ERROR
 C CRITERION IN LEGENDRE EXP-
 C SION(3)
 C NOTE:IF IEK.LT.0 SUPPRESS ANALYTIC DETERMINATION OF
 C OF FIRST COLLIDED FLUX WHEN APPROPRIATE
 C
 C LM = NUMBER OF CYCLES FOR ACCELERATED
 C CONVERGENCE(3)
 C
 C CARD 9
 C ERK = RELATIVE TRUNCATION ERROR FOR
 C LEGENDRE EXPANSION(1.0E-04)
 C
 C CARD 10
 C ITR --1 PRINT OUTPUT ON PRINTER (132 COLUMNS)
 C = 1 PRINT OUTPUT ON TERMINAL (72 COLUMNS)
 C
 C IP --1 DO NOT PRINT INPUT EDIT
 C = 0 PRINT INPUT EDIT WITH OUTPUT EDIT(TAPE21)
 C = 1 PRINT INPUT EDIT ON TAPE23
 C
 C IXY --1 S ON ABCISSA,X ON ORDINATE IN EDIT
 C =1 X ON ABCISSA,S ON ORDINATE IN EDIT
 C
 C IPL --1 NO PLOTS REQUESTED
 C = 1 WRITE PLOT FILE(TAPE22)
 C
 C CARD 11
 C ITPE --1 READ FNK FROM TAPE24 AND COMPLETE
 C CALCULATION
 C = 0 CREATE FNK ON TAPE24 AND COMPLETE
 C CALCULATION
 C = 1 CREATE FNK ON TAPE24 AND STOP
 C
 C CARD 12 IF ITC=1
 C FTC = SCATTERING COEFFICIENTS (IC=0,LL)
 C
 C CARD 13A IF JT=-1
 C SS(IT)= PATH GRID POINTS(IT=1,LS)
 C
 C CARD 13B IF JT=1
 C S0 = INITIAL PATH LENGTH

```
C      DS = PATH INTERVAL
C
C      CARD 14A IF JX=-1
C      XX(IX)= SPACE GRID POINTS (IX=1,LX)
C
C      CARD 14B IF JX=1
C      X0 = INITIAL SPACE GRID POINT
C      DX = SPACE MESH SPACING
C
C*****
```

- 3) Screened-Rutherford transport corrected scattering coefficients
- 4) Power law variation of λ^{*-1}
- 5) Comparison with diffusion theory.

The flux as determined by transport theory can be compared to the following diffusion theory approximations:

- a) isotropic source at $x = 0$

$$\phi^D(x,t) = Q_0 \frac{e^{-x^2/4Ds}}{\sqrt{4\pi Ds}} \quad (23a)$$

$$D = \frac{1}{3(1-\omega_1)\lambda^*}$$

- b) beam source at $x = 0$

$$\phi^D(x,t) = \frac{e^{-x^2/4Ds}}{\sqrt{4\pi Ds}} \left[1 + \frac{3}{2} \mu_0 \eta \right] \quad (23b)$$

- c) isotropic source uniformly distributed in $x \leq 0$

$$\phi^D(x,t) = \frac{Q_0}{2} \left[1 - \text{erf} (x/\sqrt{4Ds}) \right] \quad (24a)$$

- d) beam source uniformly distributed in $x \leq 0$

$$\phi^D(x,t) = \frac{Q_0}{2} \left\{ [1 - \text{erf} (x/\sqrt{4Ds})] + 3 \mu_0 \sqrt{\frac{D}{\pi s}} e^{-x^2/4Ds} \right\}. \quad (24b)$$

- 6) Restart - Once the expansion coefficients $f_{n,k}^\ell$ have been determined, they can be used again in calculations where the angular source distribution and order of scattering remain unchanged. For this reason, the coefficients $f_{n,k}^\ell$ are written on file 04 and are available.

Use of this option considerably reduces the computational effort.

B. Sample Problems

The sample problems to follow are presented to illustrate the features of the SLEET code and to demonstrate its use.

1) Problem 1

The first problem is the basic isotropic scattering and source problem where the source is at $x = 0$. The flux is determined at $x = 1(1)5$ for paths $S = 1(1)20$.

Along with the fluxes, diagnostic information indicating the convergence properties of the reconstruction at each time and space point is given in the output edit. The information is contained in coded number under the title "NUMBER OF TERMS FOR CONVERGENCE". Each entry corresponds to the x and s in the table above. If no infinite series evaluations were performed for a particular x, t point, then a zero entry is found. This will be the case in the forbidden region ($s < x$) or if x and s are equal and the flux is the uncollided contribution only. Otherwise a number of the form $bxx.yyyuE\pm zz$ will be found. The number is coded as follows:

u	is the number of convergence accelerations required
yyy	is the maximum number of terms required for convergence of the multiple collision series
-zz	indicates the Legendre series has converged to the desired relative error($ERK/10$) with zz being the estimated relative error exponent for the multiple collision series
+zz	indicates the Legendre series did not converge to the desired relative error with zz now representing the exponent of the minimum relative error for the Legendre series
b=-	indicates that more than one convergence iteration (IV.B) has been performed in order to achieve the desired precision
b=blank	indicates that no convergence iterations were necessary to achieve the desired precision
xx	represents the number of terms required for convergence of the Legendre series unless a + sign precedes zz . In this case, the series did not converge and xx is the term in the series producing the least relative error. The flux has been set equal to the partial sum up to and including this term.

a) Input: File SLEET.DAT

SAMPLE PROBLEM 1: ISOTROPIC SOURCE AT X=0, ISOTROPIC MEDIUM

1 0 1.0 1.0

0 1 1

1.0

0.0 1.0

20 5 1 1

1

90 60 3 3

1.0E-04

1 1 1 -1

0

1.0

1.0 1.0

1.0 1.0

b) Input Echo: File 03

SAMPLE PROBLEM 1:ISOTROPIC SOURCE AT X=0,ISOTROPIC MEDIUM

IS = 1 1=ISOTROPIC SOURCE
 2=ANISOTROPIC SOURCE

JTI= 0 0=DELTA FUNCTION SOURCE AT X=0
 1=CONSTANT SOURCE IN LEFT HALF-SPACE(X<0)

Q0 = 1.0000E+00 SOURCE STRENGTH

ALA = 1.0000E+00 MEAN FREE PATH

LL = 0 ORDER OF SCATTERING APPROXIMATION

ITC = 1 -1=USE TEST KERNEL
 1=READ COEFFICIENTS

ITRN= 1 -1=USE TRANSPORT CORRECTED CROSS SECTIONS
 1=DO NOT USE TRANSPORT CORRECTION

SCATTERING COEFFICIENTS(L=0- 0)
1.0000E+00

BE = 0.0000E+00 EXPONENT IN VARIATION OF C
 (C(S)=C*(S/SI0)**BE)

SI0 = 1.0000E+00 PARAMETER IN VARIABLE C

LS = 20 NUMBER OF PATH POINTS

LX = 5 NUMBER OF SPACE POINTS

JT = 1 -1=READ IN PATH POINTS
 1=READ IN MESH SPACING

JX = 1 -1=READ IN SPACE POINTS
 1=READ IN MESH SPACING

S0 = 1.0000E+00 INITIAL PATH

DS = 1.0000E+00 PATH INTERVAL

X0 = 1.0000E+00 FIRST POSITION

DX = 1.0000E+00 SPACE INTERVAL

PATH POINTS

1.0000E+00	2.0000E+00	3.0000E+00	4.0000E+00	5.0000E+00
6.0000E+00	7.0000E+00	8.0000E+00	9.0000E+00	1.0000E+01
1.1000E+01	1.2000E+01	1.3000E+01	1.4000E+01	1.5000E+01
1.6000E+01	1.7000E+01	1.8000E+01	1.9000E+01	2.0000E+01

SPACE POINTS

1.0000E+00	2.0000E+00	3.0000E+00	4.0000E+00	5.0000E+00
------------	------------	------------	------------	------------

IDFF= 1 -1=DIFFUSION COMPARISON
 1=SCALAR FLUX CALCULATION

LN = 90 MAXIMUM NUMBER OF TERMS IN
 MULTIPLE COLLISION EXPANSION

ERN = 1.0000E-05 RELATIVE TRUNCATION ERROR
 FOR MULTIPLE COLLISION EXPANSION

IEN = 3 NUMBER OF TERMS SATISFYING RELATIVE

ERROR CRITERION

LK = 60	MAXIMUM NUMBER OF TERMS IN LEGENDRE EXPANSION
ERK = 1.0000E-04	RELATIVE TRUNCATION ERROR
IEK = 3	NUMBER OF TERMS SATISFYING RELATIVE ERROR CRITERION
LM = 3	NUMBER OF PA ES FOR CONVERGENCE(JTI=0)
ITR = 1	-1=OUTPUT TO PRINTER(132 COLUMNS) 1=OUTPUT TO TERMINAL(72 COLUMNS)
IP = 1	-1=DO NOT PRINT INPUT EDIT 0=PRINT INPUT EDIT ON TAPE21 1=PRINT INPUT EDIT ON TAPE23
IXY = 1	-1=S ON ABCISSA AND X ON ORDINATE IN OUTPUT EDIT 1=X ON ABCISSA AND S ON ORDINATE IN OUTPUT EDIT
IPL = -1	-1=NO PLOTS REQUESTED 1=CALL FOR OUTPUT PLOTS ON TAPE22
ITPE= 0	-1=READ FNK FROM TAPE24 AND COMPLETE CALCULATION 0=CREATE FNK ON TAPE24 AND COMPLETE CALCULATION 1=CREATE FNK IN TAPE24 AND STOP

c) Flux Output: File 01

MINIMUM CONTAINER ARRAY DIMENSION= 18324

PLANE GEOMETRY

SOURCE:

AT X=0

ISOTROPIC

SCATTERING ORDER LL= 0

LEGENDRE COEFFICIENTS:

1.00000E+00

ALA= 1.0000E+00

BE = 0.00000E+00 SI0= 1.00000E+00

FLUX CALCULATION

S/X	1.0000E+00	2.0000E+00	3.0000E+00	4.0000E+00	5.0000E+00
1.0000E+00	1.8394E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
2.0000E+00	2.6256E-01	3.3834E-02	0.0000E+00	0.0000E+00	0.0000E+00
3.0000E+00	2.3943E-01	9.3839E-02	8.2978E-03	0.0000E+00	0.0000E+00
4.0000E+00	2.1736E-01	1.1361E-01	3.3288E-02	2.2895E-03	0.0000E+00
5.0000E+00	1.9957E-01	1.2105E-01	4.9595E-02	1.1823E-02	6.7379E-04
6.0000E+00	1.8523E-01	1.2322E-01	6.0605E-02	2.0710E-02	4.2106E-03
7.0000E+00	1.7348E-01	1.2293E-01	6.8028E-02	2.8447E-02	8.4158E-03
8.0000E+00	1.6364E-01	1.2143E-01	7.3033E-02	3.4901E-02	1.2790E-02
9.0000E+00	1.5528E-01	1.1935E-01	7.6385E-02	4.0186E-02	1.7004E-02
1.0000E+01	1.4806E-01	1.1700E-01	7.8590E-02	4.4479E-02	2.0905E-02
1.1000E+01	1.4175E-01	1.1455E-01	7.9986E-02	4.7954E-02	2.4433E-02
1.2000E+01	1.3617E-01	1.1209E-01	8.0803E-02	5.0760E-02	2.7583E-02
1.3000E+01	1.3120E-01	1.0969E-01	8.1201E-02	5.3024E-02	3.0372E-02
1.4000E+01	1.2674E-01	1.0737E-01	8.1293E-02	5.4845E-02	3.2828E-02
1.5000E+01	1.2269E-01	1.0514E-01	8.1159E-02	5.6305E-02	3.4986E-02
1.6000E+01	1.1901E-01	1.0300E-01	8.0860E-02	5.7470E-02	3.6876E-02
1.7000E+01	1.1564E-01	1.0096E-01	8.0438E-02	5.8391E-02	3.8531E-02
1.8000E+01	1.1254E-01	9.9019E-02	7.9927E-02	5.9111E-02	3.9978E-02
1.9000E+01	1.0968E-01	9.7167E-02	7.9349E-02	5.9664E-02	4.1241E-02
2.0000E+01	1.0702E-01	9.5402E-02	7.8725E-02	6.0078E-02	4.2344E-02

NUMBER OF TERMS FOR CONVERGENCE

00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00
20.0180E-05	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00
18.0200E-05	20.0200E-05	00.0000E+00	00.0000E+00	00.0000E+00
18.0230E-05	18.0230E-05	22.0230E-05	00.0000E+00	00.0000E+00
20.0260E-05	20.0260E-05	22.0260E-05	22.0260E-05	00.0000E+00
20.0280E-05	20.0280E-05	20.0280E-05	22.0280E-05	24.0300E-06
20.0320E-05	22.0320E-05	22.0320E-05	24.0320E-05	24.0320E-05
18.0310E-05	22.0330E-05	24.0330E-05	24.0330E-05	22.0330E-05
20.0340E-05	24.0390E-05	24.0390E-05	22.0390E-05	24.0390E-05
22.0360E-05	24.0370E-05	24.0370E-05	24.0370E-05	26.0380E-05

22.0380E-05	24.0390E-05	22.0390E-05	26.0400E-05	26.0400E-05
22.0390E-05	26.0410E-05	24.0410E-05	26.0410E-05	24.0410E-05
24.0410E-05	26.0420E-05	26.0420E-05	26.0420E-05	26.0420E-05
24.0420E-05	26.0430E-05	26.0430E-05	28.0440E-05	26.0440E-05
26.0450E-05	26.0450E-05	26.0450E-05	28.0460E-05	28.0460E-05
26.0460E-05	28.0470E-05	28.0470E-05	28.0470E-05	28.0470E-05
26.0470E-05	28.0480E-05	28.0480E-05	26.0480E-05	30.0490E-05
28.0490E-05	28.0490E-05	28.0490E-05	26.0490E-05	30.0500E-05
28.0510E-05	28.0510E-05	30.0520E-05	28.0520E-05	30.0520E-05
28.0520E-05	28.0520E-05	30.0530E-05	28.0530E-05	30.0530E-05

2) Problem 2

In this problem a beam source emitting electrons in direction $\mu_0 = 1$ is located at $x = 0$. A transport corrected screened-Rutherford (SR) scattering kernel is assumed with $L = 35$ and $\eta = 0.05$. The same points as in problem 1 are edited. Note the appearance of the unlikely number $7.7777E+17$ when $s = x$. This value indicates the delta function "infinity" of the uncollided flux.

a) Input: File SLEET.DAT

SAMPLE PROBLEM 2: BEAM SOURCE AT X=0, SR-LL=35

2 0 1.0 1.0

35 -1 1

0.05 1.0

0.0 1.0

20 5 1 1

1

90 90 3 3

1.0E-04

1 1 1 -1

0

1.0 1.0

1.0 1.0

b) Flux Output: File 01

MINIMUM CONTAINER ARRAY DIMENSION= 26844

PLANE GEOMETRY

SOURCE:

AT X=0

ANISOTROPIC

MU0= 1.0000E+00

SCATTERING ORDER LL= 35

LEGENDRE COEFFICIENTS:

1.00000E+00	8.59679E-01	7.07990E-01	5.68955E-01	4.49953E-01
3.51820E-01	2.72761E-01	2.10075E-01	1.60941E-01	1.22766E-01
9.33080E-02	7.07013E-02	5.34309E-02	4.02871E-02	3.03158E-02
2.27721E-02	1.70787E-02	1.27907E-02	9.56708E-03	7.14768E-03
5.33453E-03	3.97750E-03	2.96308E-03	2.20559E-03	1.64052E-03
1.21938E-03	9.05761E-04	6.72402E-04	4.98886E-04	3.69953E-04
2.74206E-04	2.03145E-04	1.50435E-04	1.11355E-04	8.23959E-05
6.09456E-05				

ETA= 5.0000E-02 ALA= 1.0000E+00

BE = 0.00000E+00 SI0= 1.00000E+00

FLUX CALCULATION

S/X	1.0000E+00	2.0000E+00	3.0000E+00	4.0000E+00	5.0000E+00
1.0000E+00	7.7777E+17	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
2.0000E+00	9.9542E-02	7.7777E+17	0.0000E+00	0.0000E+00	0.0000E+00
3.0000E+00	5.9803E-02	2.0083E-01	7.7777E+17	0.0000E+00	0.0000E+00
4.0000E+00	4.9196E-02	1.0513E-01	2.8002E-01	7.7777E+17	0.0000E+00
5.0000E+00	4.4716E-02	7.7885E-02	1.4534E-01	3.1745E-01	7.7777E+17
6.0000E+00	4.2405E-02	6.5544E-02	1.0361E-01	1.7253E-01	3.1605E-01
7.0000E+00	4.1050E-02	5.8607E-02	8.3991E-02	1.2251E-01	1.8462E-01
8.0000E+00	4.0167E-02	5.4178E-02	7.2725E-02	9.7921E-02	1.3318E-01
9.0000E+00	3.9526E-02	5.1074E-02	6.5419E-02	8.3464E-02	1.0649E-01
1.0000E+01	3.9011E-02	4.8751E-02	6.0266E-02	7.3958E-02	9.0329E-02
1.1000E+01	3.8560E-02	4.6911E-02	5.6409E-02	6.7214E-02	7.9516E-02
1.2000E+01	3.8136E-02	4.5391E-02	5.3379E-02	6.2157E-02	7.1766E-02
1.3000E+01	3.7725E-02	4.4087E-02	5.0914E-02	5.8200E-02	6.5927E-02
1.4000E+01	3.7311E-02	4.2942E-02	4.8848E-02	5.4999E-02	6.1354E-02
1.5000E+01	3.6894E-02	4.1913E-02	4.7076E-02	5.2344E-02	5.7654E-02
1.6000E+01	3.6470E-02	4.0971E-02	4.5520E-02	5.0082E-02	5.4592E-02
1.7000E+01	3.6043E-02	4.0100E-02	4.4143E-02	4.8128E-02	5.2004E-02
1.8000E+01	3.5609E-02	3.9287E-02	4.2904E-02	4.6416E-02	4.9779E-02
1.9000E+01	3.5174E-02	3.8521E-02	4.1776E-02	4.4897E-02	4.7839E-02
2.0000E+01	3.4737E-02	3.7797E-02	4.0742E-02	4.3533E-02	4.6129E-02

NUMBER OF TERMS FOR CONVERGENCE

00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00
-53.0131E-05	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00

-50.0161E-05	-44.0161E-05	00.0000E+00	00.0000E+00	00.0000E+00
-46.0181E-05	-40.0181E-05	38.0181E-05	00.0000E+00	00.0000E+00
-39.0201E-05	-37.0201E-05	35.0201E-05	34.0201E-05	00.0000E+00
-35.0221E-05	32.0221E-05	31.0221E-05	31.0221E-05	73.0220E-05
-30.0251E-05	29.0251E-05	86.0250E-05	77.0250E-05	64.0250E-05
80.0270E-05	26.0270E-05	26.0270E-05	26.0270E-05	26.0270E-05
34.0290E-05	34.0290E-05	33.0290E-05	33.0290E-05	33.0290E-05
34.0300E-05	34.0300E-05	34.0300E-05	34.0300E-05	32.0300E-05
32.0330E-05	31.0330E-05	31.0330E-05	31.0330E-05	30.0330E-05
30.0340E-05	29.0340E-05	30.0340E-05	30.0340E-05	29.0340E-05
27.0390E-05	28.0390E-05	28.0390E-05	27.0390E-05	26.0390E-05
29.0380E-05	26.0380E-05	25.0380E-05	24.0380E-05	25.0380E-05
29.0390E-05	30.0390E-05	27.0390E-05	23.0390E-05	25.0390E-05
30.0400E-05	30.0400E-05	28.0400E-05	28.0400E-05	29.0400E-05
31.0430E-05	30.0430E-05	27.0430E-05	30.0430E-05	28.0430E-05
29.0430E-05	28.0430E-05	29.0430E-05	28.0430E-05	28.0430E-05
27.0450E-05	28.0450E-05	26.0450E-05	27.0450E-05	26.0450E-05
27.0470E-05	26.0470E-05	26.0470E-05	27.0470E-05	26.0470E-05

3) Problem 3

The final problem is identical to problem 2 but the source is located in the entire left half-space ($x < 0$).

a) Input: File SLEET.DAT

SAMPLE PROBLEM 3: BEAM SOURCE IN A HALF-SPACE, SR-LL-35

2 1 1.0 1.0

35 -1 1

0.05 1.0

0.0 1.0

5 22 1 1

1

75 75 3 3

1.0E-04

1 1 -1 -1

0

1.0 2.0

-11.0 1.0

MINIMUM CONTAINER ARRAY DIMENSION= 19018

PLANE GEOMETRY

SOURCE:

CONSTANT HALF-SPACE

ANISOTROPIC

MU0= 1.0000E+00

SCATTERING ORDER LL= 35

LEGENDRE COEFFICIENTS:

1.00000E+00	8.59679E-01	7.07990E-01	5.68955E-01	4.49953E-01
3.51820E-01	2.72761E-01	2.10075E-01	1.60941E-01	1.22766E-01
9.33080E-02	7.07013E-02	5.34309E-02	4.02871E-02	3.03158E-02
2.27721E-02	1.70787E-02	1.27907E-02	9.56708E-03	7.14768E-03
5.33453E-03	3.97750E-03	2.96308E-03	2.20559E-03	1.64052E-03
1.21938E-03	9.05761E-04	6.72402E-04	4.98886E-04	3.69953E-04
2.74206E-04	2.03145E-04	1.50435E-04	1.11355E-04	8.23959E-05
6.09456E-05				

ETA= 5.0000E-02 ALA= 1.0000E+00

BE = 0.00000E+00 SIO= 1.00000E+00

FLUX CALCULATION

X/S	1.0000E+00	3.0000E+00	5.0000E+00	7.0000E+00	9.0000E+00

-1.1000E+01	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00
-1.0000E+01	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00
-9.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00
-8.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	9.9995E-01
-7.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	9.9947E-01
6.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	9.9987E-01	9.9794E-01
-5.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	9.9886E-01	9.9467E-01
-4.0000E+00	1.0000E+00	1.0000E+00	9.9962E-01	9.9600E-01	9.8884E-01
-3.0000E+00	1.0000E+00	1.0000E+00	9.9746E-01	9.9023E-01	9.7961E-01
-2.0000E+00	1.0000E+00	9.9894E-01	9.9184E-01	9.8029E-01	9.6591E-01
-1.0000E+00	1.0000E+00	9.9352E-01	9.8062E-01	9.6451E-01	9.4663E-01
0.0000E+00	9.9354E-01	9.7833E-01	9.6043E-01	9.4087E-01	9.2044E-01
1.0000E+00	3.6788E-01	9.3998E-01	9.2579E-01	9.0641E-01	8.8577E-01
2.0000E+00	0.0000E+00	8.2800E-01	8.6627E-01	8.5712E-01	8.4066E-01
3.0000E+00	0.0000E+00	4.9787E-02	7.5907E-01	7.8660E-01	7.8269E-01
4.0000E+00	0.0000E+00	0.0000E+00	5.4290E-01	6.8482E-01	7.0860E-01
5.0000E+00	0.0000E+00	0.0000E+00	6.7379E-03	5.3385E-01	6.1410E-01
6.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	3.0161E-01	4.9344E-01
7.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	9.1188E-04	3.3960E-01
8.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.5027E-01
9.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.2341E-04
1.0000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

NUMBER OF TERMS FOR CONVERGENCE

00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00
00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00
00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00
00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	20.0290E-05
00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	20.0290E-05
00.0000E+00	00.0000E+00	00.0000E+00	22.0260E-05	20.0290E-05
00.0000E+00	00.0000E+00	00.0000E+00	22.0260E-05	19.0290E-05
00.0000E+00	00.0000E+00	22.0200E-05	22.0260E-05	19.0290E-05
00.0000E+00	00.0000E+00	23.0200E-05	22.0260E-05	18.0290E-05
00.0000E+00	23.0150E-05	24.0200E-05	22.0260E-05	19.0290E-05
00.0000E+00	23.0150E-05	23.0200E-05	21.0260E-05	20.0290E-05
37.0100E-05	23.0150E-05	23.0200E-05	21.0260E-05	19.0290E-05
00.0000E+00	23.0150E-05	23.0200E-05	21.0260E-05	20.0290E-05
00.0000E+00	24.0150E-05	24.0200E-05	22.0260E-05	19.0290E-05
00.0000E+00	00.0000E+00	23.0200E-05	22.0260E-05	20.0290E-05
00.0000E+00	00.0000E+00	26.0200E-05	22.0260E-05	20.0290E-05
00.0000E+00	00.0000E+00	00.0000E+00	23.0260E-05	19.0290E-05
00.0000E+00	00.0000E+00	00.0000E+00	-25.0260E-05	21.0290E-05
00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	21.0290E-05
00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	-21.0290E-05
00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00
00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00	00.0000E+00

C) Programming Notes

The SLEET code is structured on a modular basis. The main routine calls the input routines and for the calculation or input from tape (TAPE 24) of $f_{n,k}^0$ as well as for an edit of the input options (on TAPE 21 or 23) if desired. In addition, the path is advanced and a call for the flux determination (in SCALR) is made along with the output edit. The use of dynamic storage allows maximum flexibility in the problem size. A container array size is located in a parameter statement in the main program. If $IASK > IMX$ where IMX is the maximum storage requirement for the given problem, a diagnostic message is printed on the screen and the calculation is terminated. The minimum container array required is also printed in the output (TAPE 21) for use in the sizing of problems.

REFERENCES

1. H. W. Lewis, Phys. Rev., 78, 526 (1950).
2. L. V. Spencer, Phys. Rev., 98, 1597 (1955).
3. D. E. Bartine, R. G. Alsmiller, Jr., F. R. Mynatt, W. W. Engle, Jr. and J. Barish, NASA TMX-2440, 816 (1972).
4. W. J. Wiscombe, Jour. of Atmos. Sci., 34, 1408 (1977).
5. S. A. Kholin, USSR Comp. Math. and Math Phys., 4, 213 (1964).

APPENDIX

Derivation of Legendre Scattering Coefficients for the Screened Rutherford Scattering Kernel.

The derivation begins with the screened-Rutherford scattering law modified to incorporate a localized electron-atom collision (screening $\eta > 0$):

$$g(\mu_0) = \frac{1}{2\pi} \frac{\eta(1+\eta/2)}{(1+\eta - \mu_0)^2} \quad (A.1)$$

If g is expanded in terms of Legendre polynomials, we have

$$g(\mu_0) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \omega_{\ell} P_{\ell}(\mu_0)$$

with

$$\omega_{\ell} = 2\pi \int_{-1}^1 d\mu_0 P_{\ell}(\mu_0) g(\mu_0) \quad (A.2)$$

and upon substitution

$$\omega_{\ell} = \frac{\eta}{\pi} (1+\eta/2) \frac{1}{2} \int_{-1}^1 \frac{d\mu_0 P_{\ell}(\mu_0)}{(z-\mu_0)^2}, \quad z = 1+\eta. \quad (A.3)$$

By noting that

$$Q_{\ell}(z) = \frac{1}{2} \int_{-1}^1 d\mu_0 \frac{P_{\ell}(\mu_0)}{z-\mu_0}$$

and

$$\frac{dQ_{\ell}(z)}{dz} = - \frac{1}{2} \int_{-1}^1 d\mu \frac{P_{\ell}(\mu_0)}{(z-\mu_0)^2},$$

Eq. (A.3) becomes

$$\omega_{\ell} = \frac{\eta}{\pi} \left(1 + \frac{\eta}{2}\right) \frac{dQ_{\ell}(z)}{dz} \quad (A.4)$$

Since

$$\frac{dQ_{\ell}}{dz} = - \frac{Q_{\ell}^1(z)}{\sqrt{1-z^2}}$$

where Q_{ℓ}^1 is an associated Legendre function of the second kind of order ℓ satisfying

$$\ell Q_{\ell-1}^1 = (2\ell+1)(1+\hat{\eta}) Q_{\ell}^1 - (\ell+1) Q_{\ell-1}^1, \quad (A.5)$$

we can write for Eq. (A.4)

$$\omega_{\ell} = \frac{\hat{\eta}}{\pi} \left(1 + \frac{\hat{\eta}}{2}\right) \frac{Q_{\ell}^1(z)}{\sqrt{1-z^2}} . \quad (\text{A.6})$$

From Eq. (A.5) therefore,

$$\ell \omega_{\ell+1} = (2\ell+1)(1+\hat{\eta}) \omega_{\ell} - (\ell+1) \omega_{\ell-1} . \quad (\text{A.7})$$

This recurrence relation is initiated by ω_0, ω_1 given by Eqs. (5a,b) which are obtained directly from Eq. (A.2).

PROGRAM SLEET

C****

C MAIN PROGRAM

C CHOOSES INPUT OPTIONS

C ADVANCES PATH COORDINATE

C CHOOSES OUTPUT EDIT OPTIONS

C****

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

PARAMETER (IASK=50000)

COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LX,JT,JX,

* IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,

* KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT

COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SIO,ERN,ERK,DC,S0,DS,X0,DX

COMMON/C3/PI,X,S,Q,QY,TX,SX1,ET,TS,PH1,FD,DPH,WLP1

COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX

DIMENSION A(80000)

OPEN(UNIT=20,FILE='SLEETM.DAT',STATUS='OLD')

OPEN(UNIT=21,FILE='01',STATUS='UNKNOWN')

OPEN(UNIT=22,FILE='02',STATUS='UNKNOWN')

OPEN(UNIT=23,FILE='03',STATUS='UNKNOWN')

OPEN(UNIT=24,FILE='04',STATUS='UNKNOWN')

C*** DIMENSION OF CONTAINER ARRAY

MLT=3

C***

C SET CONSTANTS AND ARRAY POINTERS

C*CDC EXMX=199

EXMX=80

IN=20

IOT1=21

IOT2=22

IOT3=23

IOT4=24

IOT06=6

REWIND IOT1

REWIND IOT2

REWIND IOT3

REWIND IOT4

PI=4.0D0*DATAN(1.0D0)

IR=4

WRITE(IOT06,*) 'READ INPUT AND SET ARRAY POINTERS'

CALL INPUT

LA=LS*MX(1)+LX*MX(2)

LO=LS*MY(1)+LX*MY(2)

LNR=IR+LN

LKR=IR+LK

LLR=IR+LL

LMR=LKR+MLT

IXX=1

ISS=IXX+LX

IFTC=ISS+LS

IAB=IFTC+LKR+MLT

IOR=IAB+LA

IDT=IOR+LO

IMXT=IDT+LLR

DO 1 I=1,IMXT

A(I)=0.0

```

1 CONTINUE
  CALL INPUT1(A(IXX),A(ISS),A(IFTC),A(IAB),A(IOR)
  * ,LX,LS,LA,LO,LLR,LMR)
  DO 2 I=IDT,IASK
    A(I)=0.0D0
2 CONTINUE
  IGN=IDT
  IPO=IGN+LKR*LNR
  IFNT=IPO+LKR
  IF(ITPE.LT.0) GO TO 103
  IHNT=IFNT+LNR*LKR
  IWT=IHNT+LNR*LKR
  IMX=IWT+LNR
  IMX1=IMX
  IF(IMX.LE.IASK) GO TO 150
  WRITE(IOT06,199) IASK,IMX
  WRITE(IOT1,199) IASK,IMX
199 FORMAT(1X,'INSUFFICIENT STORAGE IASK=',I6,2X,'NEED=',I6,' FOR COMP
  *LETION')
  STOP
150 CONTINUE
  WRITE(IOT06,*) 'DETERMINE FLUX EXPANSION COEFFICIENTS'
  CALL ALP4(A(IFNT),A(IGN),A(IHNT),A(IWT),A(IFTC),A(IPO)
  * ,LNR,LKR)
  WRITE(IOT06,*) 'EXPANSION COEFFICIENTS STORED ON FILE 02'
  IF(IP.LT.0) GO TO 112
  IF(IP.EQ.0) IOT3=21
  IF(IP.EQ.1)
  *WRITE(IOT06,*) 'INPUT WRITTEN ON FILE 03'
  IF(IP.EQ.0)
  *WRITE(IOT06,*) 'INPUT WRITTEN ON FILE 01'
  CALL INPRNT(A(IXX),A(ISS),A(IFTC),LX,LS,LLR)
  IP=-1
112 CONTINUE
  IF(ITPE.NE.0) STOP
103 CONTINUE
  IF(IP.LT.0) GO TO 111
  IF(IP.EQ.0) IOT3=21
  IF(IP.EQ.1)
  *WRITE(IOT06,*) 'INPUT WRITTEN ON FILE 03'
  IF(IP.EQ.0)
  *WRITE(IOT06,*) 'INPUT WRITTEN ON FILE 01'
  CALL INPRNT(A(IXX),A(ISS),A(IFTC),LX,LS,LLR)
111 CONTINUE
  DO 3 I=IGN,IASK
    A(I)=0.0D0
3 CONTINUE
  IPHI=IFNT
  IERR=IPHI+LA*LO
  IQ=IERR+LA*LO
  IU=IQ+LKR
  IMX=IU+2*LKR
  IMX2=IMX
  IMXM=IMX1
  IF(IMX2.GT.IMX1) IMXM=IMX2
  IF(IMX.LE.IASK) GO TO 200

```

```

WRITE(IOT1,199) IASK,IMX
STOP
200 CONTINUE
LAP=LA
LOP=LO
DO 105 KST=1,LS
IST=KST
S=A(ISS+KST-1)
WRITE(IOT06,*) 'BEGIN S=',S
SX1=JTI*(S-1.0D0)+1.0D0
Q=S/ALA
QY=(S/SI0)**BE
QB=Q*QY/B1
IF(QB.LT.EXMX) GO TO 133
LAP=(KST-1)*MX(1)+LX*MX(2)
LOP=(KST-1)*MY(1)+LX*MY(2)
GO TO 104
133 CONTINUE
TS=DEXP(-QB)
109 CONTINUE
IF(S.EQ.0.0) GO TO 105
TX=TS/S
CALL SCALR(A(IXX),A(IGN),A(IPHI),A(IERR),A(IPO),A(IQ)
*,A(IFTC),A(IU),LNR,LKR,LA,LO,LX,LLR)
105 CONTINUE
104 CONTINUE
WRITE(IOT06,*) 'WRITE OUTPUT ON FILE 01'
CALL OUT(A(IAB),A(IOR),A(IFTC),A(IPHI),A(IERR),LA,LO,LLR)
STOP
END
SUBROUTINE INPUT
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C READ AND PREPARES INPUT
C****
C INPUT ROUTINE
C
C FREE FORMAT
C
C SUGGESTED VALUES ARE GIVE IN PARENTHESIS
C****
COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LX,JT,JX,
* IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
* KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SI0,ERN,ERK,DC,S0,DS,X0,DX
COMMON/C3/PI,X,S,Q,QY,TX,SX1,ET,TS,PH1,FD,DPH,WLP1
COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX
C*****
C INPUT DESCRIPTION
C
C CARD 1 (18A4)
C IDEN = PROBLEM IDENTIFICATION
C
C CARD 2
C IS = 1 ISOTROPIC SOURCE
C      = 2 ANISOTROPIC SOURCE

```

C JTI = 0 DELTA FUNCTION SOURCE AT
C X=0
C = 1 CONSTANT SOURCE IN LEFT
C HALF-SPACE(X<0)
C
C Q0 = SOURCE STRENGTH
C
C AM0 = COSINE OF ANGLE OF INCIDENCE OF SOURCE
C ELECTRONS
C
C CARD 3
C LL = ORDER OF SCATTERING KERNEL APPROXIMATION
C
C ITC = -1 SCREENED-RUTHERFORD SCATTERING KERNEL
C = 1 READ IN SCATTERING COEFFICIENTS
C
C ITRN = -1 USE TRANSPORT CORRECTED CROSS SECTIONS
C = 1 DO NOT USE TRANSPORT CORRECTED CROSS
C SECTIONS
C
C CARD 4A IF ITC=1
C ALA = TOTAL MEAN FREE PATH
C
C CARD 4B IF ITC=-1
C ETA = SCREENING PARAMETER FOR SCREENED
C RUTHERFORD SCATTERING CROSS SECTION
C
C ALA = TOTAL MEAN FREE PATH
C
C
C CARD 5
C BE = POWER OF PATH DEPENDENCE OF NUMBER
C OF SECONDARIES (ALA(S)=(S/SI0)**BE)
C SI0 = CROSS SECTION PARAMETER
C
C CARD 6
C LS = NUMBER OF PATH GRID POINTS
C
C LX = NUMBER OF SPACE GRID POINTS
C
C JT = -1 READ IN PATH GRID POINTS
C = 1 READ IN PATH MESH SPACING
C
C JX = -1 READ IN SPACE GRID POINTS
C = 1 READ IN SPACE MESH SPACING
C
C CARD 7
C IDFF=-1 COMPARISON WITH DIFFUSION THEORY
C (BE=0 ONLY)
C = 1 SCALAR FLUX CALCULATION
C
C CARD 8
C LN = MAXIMUM NUMBER OF TERMS IN
C MULTIPLE COLLISION EXPANSION
C OF MOMENTS
C

```

C      LK = MAXIMUM NUMBER OF TERMS IN
C      LEGENDRE SERIES EXPANSION
C
C      IEK = NUMBER OF CONSECUTIVE TERMS
C      SATISFYING RELATIVE ERROR
C      CRITERION IN LEGENDRE EXP-
C      SION(3)
C NOTE:IF IEK.LT.0 SUPPRESS ANALYTIC DETERMINATION OF
C      OF FIRST COLLIDED FLUX WHEN APPROPRIATE
C
C      LM = NUMBER OF CYCLES FOR ACCELERATED
C      CONVERGENCE(3)
C
C CARD 9
C      ERK = RELATIVE TRUNCATION ERROR FOR
C      LEGENDRE EXPANSION(1.0E-04)
C
C CARD 10
C      ITR --1 PRINT OUTPUT ON PRINTER (132 COLUMNS)
C      = 1 PRINT OUTPUT ON TERMINAL (72 COLUMNS)
C
C      IP --1 DO NOT PRINT INPUT EDIT
C      = 0 PRINT INPUT EDIT WITH OUTPUT EDIT(TAPE21)
C      = 1 PRINT INPUT EDIT ON TAPE23
C
C      IXY --1 S ON ABCISSA,X ON ORDINATE IN EDIT
C      =1 X ON ABCISSA,S ON ORDINATE IN EDIT
C
C      IPL --1 NO PLOTS REQUESTED
C      = 1 WRITE PLOT FILE(TAPE22)
C
C CARD 11
C      ITPE --1 READ FNK FROM TAPE24 AND COMPLETE
C      CALCULATION
C      = 0 CREATE FNK ON TAPE24 AND COMPLETE
C      CALCULATION
C      = 1 CREATE FNK ON TAPE24 AND STOP
C
C CARD 12 IF ITC=1
C      FTC = SCATTERING COEFFICIENTS (IC=0,LL)
C
C CARD 13A IF JT=-1
C SS(IT)= PATH GRID POINTS(IT=1,LS)
C
C CARD 13B IF JT=1
C      S0 = INITIAL PATH LENGTH
C      DS = PATH INTERVAL
C
C CARD 14A IF JX=-1
C      XX(IX)= SPACE GRID POINTS (IX=1,LX)
C
C CARD 14B IF JX=1
C      X0 = INITIAL SPACE GRID POINT
C      DX = SPACE MESH SPACING
C
C *****

```

```

C  READ INPUT
  READ(IN,3) IDEN
  3  FORMAT(18A4)
  READ(IN,*) IS,JTI,Q0,AM0
  READ(IN,*) LL,ITC,ITRN
  LLP1=LL+1
  LLP2=LL+2
  IF(ITC.LT.0) READ(IN,*) ETA,ALA
  IF(ITC.GT.0) READ(IN,*) ALA
  READ(IN,*) BE,SIO
  IF(BE.EQ.0.0) SIO=1.0D0
  B1=1.0D0+BE
  KFF=2
  KII=2
  NS=1
  IF(IS.EQ.1) GO TO 106
  KFF=1
  KII=1
  IF(AM0.NE.0.0) GO TO 104
  KFF=2
  KII=2
104  CONTINUE
  NS=2
  IF(JTI.GT.0) NS=1
  IF(BE.NE.0.0) NS=1
106  CONTINUE
  READ(IN,*) LS,LX,JT,JX
  READ(IN,*) IDFF
  READ(IN,*) LN,LK,IEK,LM
  IF(IEK.LT.0) NS=1
  IEK=IABS(IEK)
  LNP=LN+1
  IF(LL.GT.LK) LK=LL
  LKP=LK+1
  READ(IN,*) ERK
  ERN=ERK/10.0D0
  IEN=IEK
  READ(IN,*) ITR,IP,IXY,IPL
  READ(IN,*) ITPE
  DO 70 J=1,2
  MY(J)=0
  MX(J)=0
  70  CONTINUE
  IF(IXY.GT.0) GO TO 699
  MX(1)=1
  MY(2)=1
  GO TO 688
699  CONTINUE
  MX(2)=1
  MY(1)=1
688  CONTINUE
C  EXCLUSIONS
  IF(JTI.EQ.1) LM=0
  IF(LL.EQ.0) ITRN=1
  IF(BE.NE.0.0D0) IDFF=1
  RETURN

```

```

      END
      SUBROUTINE INPUT1(XX,SS,FTC,AB,OR,LX,LS,LA,LO,LLR,LMR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     READ AND PREPARE FLOATING POINT INPUT
C****
C INPUT ROUTINE
C
C     FREE FORMAT
C
C     READ AND PREPARE FLOATING POINT INPUT
C****
      COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LSS,LXX,JT,JX,
      * IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
      * KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
      COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,S10,ERN,ERK,DC,S0,DS,X0,DX
      COMMON/C3/PI,X,S,Q,QY,TX,SX1,ET,TS,PH1,FD,DPH,WLP1
      DIMENSION XX(LX),SS(LS),FTC(LLR),AB(LA),OR(LO)
      WLP1=0.0D0
      IF(ITC.LT.0) GO TO 90
      READ(IN,*) (FTC(IC),IC=IR,LLR)
      GO TO 91
90 CONTINUE
      CALL SRUTH(FTC,LMR)
91 CONTINUE
      IF(ITRN.LT.0.0) CALL TRNSC(FTC,LLR)
      F1=0.0D0
      IF(LL.GT.0) F1=FTC(IR+1)
      IF(F1.NE.1.0) DC=ALA/(3.0D0*(1-F1))
      IF(JT.GT.0) GO TO 10
      READ(IN,*) (SS(IT),IT=1,LS)
      GO TO 20
10 READ(IN,*) S0,DS
20 IF(JX.GT.0) GO TO 11
      READ(IN,*) (XX(IX),IX=1,LX)
      GO TO 21
11 CONTINUE
      READ(IN,*) X0,DX
21 IF(JT.LE.0) GO TO 60
      SS(1)=S0
      IF(LS.EQ.1) GO TO 60
      DO 61 KST=2,LS
      SS(KST)=SS(KST-1)+DS
61 CONTINUE
60 CONTINUE
      IF(JX.LE.0) GO TO 62
      XX(1)=X0
      IF(LX.EQ.1) GO TO 62
      DO 63 IX=2,LX
      XX(IX)=XX(IX-1)+DX
63 CONTINUE
62 CONTINUE
      DO 50 IX=1,LX
      IF(IXY.LT.0) OR(IX)=XX(IX)
      IF(IXY.GT.0) AB(IX)=XX(IX)
50 CONTINUE
      DO 51 IT=1,LS

```

```

      IF(IXY.LT.0) AB(IT)=SS(IT)
      IF(IXY.GT.0) OR(IT)=SS(IT)
51  CONTINUE
      RETURN
      END
      SUBROUTINE INPRNT(XX,SS,FTC,LX,LS,LLR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C****
C  INPUT PRINT ROUTINE
C  WRITES INPUT EITHER ON TAPE21
C  OR TAPE23
C****
      COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IFM,LSS,LXX,JT,JX,
      * IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
      * KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
      COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SIO,ERN,ERK,DC,S0,DS,X0,DX
      COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX
      DIMENSION XX(LX),SS(LS),FTC(LLR)
3  FORMAT(18A4)
      WRITE(IOT3,3) IDEN
      WRITE(IOT3,103) IS
103  FORMAT(/6X,'IS =',I2,14X,'1-ISOTROPIC SOURCE'/
      * 6X,20X,'2-BEAM SOURCE')
      IF(IS.EQ.2) WRITE(IOT3,121) AM0
121  FORMAT(/5X,'AM0 =',1PE12.4,6X,'COSINE OF ANGLE OF INCIDENCE'/
      + 28X,'SOURCE PARTICLES')
      WRITE(IOT3,200) JTI
200  FORMAT(/6X,'JTI=',I2,14X,'0-DELTA FUNCTION SOURCE AT X=0'/
      + 26X,'1-CONSTANT SOURCE IN LEFT HALF-SPACE(X<0)')
      WRITE(IOT3,201) Q0
201  FORMAT(/6X,'Q0 =',1PE12.4,6X,'SOURCE STRENGTH')
      IF(ITC.GT.0) GO TO 199
      WRITE(IOT3,204) ETA,ALA
204  FORMAT(/5X,'ETA =',1PE12.4,6X,'SCREENING PARAMETER'/
      * 5X,'ALA=',1PE12.4,6X,'MEAN FREE PATH')
      GO TO 198
199  CONTINUE
      WRITE(IOT3,197) ALA
197  FORMAT(/5X,'ALA =',1PE12.4,6X,'MEAN FREE PATH')
198  CONTINUE
      WRITE(IOT3,202) LL
202  FORMAT(/6X,'LL =',I2,16X,'ORDER OF SCATTERING APPROXIMATION')
      WRITE(IOT3,208) ITC,ITRN
208  FORMAT(/5X,'ITC =',I2,12X,' -1-USE TEST KERNEL'/
      + 25X,' 1-READ COEFFICIENTS'/
      + 5X,'ITRN=',I2,12X,' -1-USE TRANSPORT CORRECTED CROSS SECTIONS'/
      + 25X,' 1-DO NOT USE TRANSPORT CORRECTION')
      WRITE(IOT3,203) LL,(FTC(IC),IC=IR,LLR)
203  FORMAT(/6X,'SCATTERING COEFFICIENTS(L=0-',I2,')'/
      + (1X,5(1PE12.4)))
      WRITE(IOT3,207) BE,SIO
207  FORMAT(/6X,'BE =',1PE12.4,6X,'EXPONENT IN VARIATION OF C'/
      + 28X,20H(C(S)=C*(S/SIO)**BE)/
      + 5X,'SIO =',1PE12.4,6X,'PARAMETER IN VARIABLE C')
      WRITE(IOT3,106) LS,LX
106  FORMAT(/6X,'LS =',I3,15X,'NUMBER OF PATH POINTS'/

```

```

* 6X,'LX =',I3,15X,'NUMBER OF SPACE POINTS')
  WRITE(IOT3,105) JT,JX
105 FORMAT(/6X,'JT =',I2,12X,' -1-READ IN PATH POINTS'/
* 24X,' 1-READ IN MESH SPACING'/
* 6X,'JX =',I2,12X,' -1-READ IN SPACE POINTS'/
* 24X,' 1-READ IN MESH SPACING')
  IF(JT.GT.0)
+ WRITE(IOT3,117) S0,DS
117 FORMAT(/6X,'S0 =',1PE12.4,6X,'INITIAL PATH'/
* 6X,'DS =',1PE12.4,6X,'PATH INTERVAL')
  IF(JX.GT.0)
+ WRITE(IOT3,118) X0,DX
118 FORMAT(6X,'X0 =',1PE12.4,6X,'FIRST POSITION'/
* 6X,'DX =',1PE12.4,6X,'SPACE INTERVAL')
  WRITE(IOT3,107) (SS(IT),IT=1,LS)
107 FORMAT(/6X,'PATH POINTS'/(1X,5(1PE12.4)))
  WRITE(IOT3,108) (XX(IX),IX=1,LX)
108 FORMAT(/6X,'SPACE POINTS'/(1X,5(1PE12.4)))
  WRITE(IOT3,300) IDFF
300 FORMAT(/5X,'IDFF =',I2,12X,' -1-DIFFUSION COMPARISON'/
+ 25X,' 1-SCALAR FLUX CALCULATION')
  WRITE(IOT3,303) LN
303 FORMAT(/6X,'LN =',I3,15X,'MAXIMUM NUMBER OF TERMS IN'/
+ 28X,'MULTIPLE COLLISION EXPANSION')
  WRITE(IOT3,304) ERN,IEN
304 FORMAT(5X,'ERN =',1PE12.4,6X,'RELATIVE TRUNCATION ERROR'/
+ 28X,'FOR MULTIPLE COLLISION EXPANSION'/
+ 5X,'IEN =',I2,16X,'NUMBER OF TERMS SATISFYING RELATIVE'/
+ 28X,'ERROR CRITERION')
  WRITE(IOT3,110) LK
110 FORMAT(/6X,'LK =',I3,15X,'MAXIMUM NUMBER OF TERMS IN'/
+ 28X,'LEGENDRE EXPANSION')
  WRITE(IOT3,111) ERK,IEK,LM
111 FORMAT(5X,'ERK =',1PE12.4,6X,'RELATIVE TRUNCATION ERROR'/
* 5X,'IEK =',I2,16X,'NUMBER OF TERMS SATISFYING RELATIVE'/
* 28X,'ERROR CRITERION'/
* 5X,' LM =',I2,16X,'NUMBER OF PATTES FOR CONVERGENCE(JTI=0)
*')
  WRITE(IOT3,116) ITR,IP
116 FORMAT(/5X,'ITR =',I2,12X,' -1-OUTPUT TO PRINTER(132 COLUMNS)'/
* 26X,'1-OUTPUT TO TERMINAL(72 COLUMNS)'/
*/6X,'IP =',I2,12X,' -1-DO NOT PRINT INPUT EDIT'/
* 25X,' 0-PRINT INPUT EDIT ON TAPE21'/
* 26X,'1-PRINT INPUT EDIT ON TAPE23')
  WRITE(IOT3,305) IXY
305 FORMAT(/5X,'IXY =',I2,12X,' -1-S ON ABCISSA AND X ON'/
+ 28X,' ORDINATE IN OUTPUT EDIT'/
+ 25X,' 1-X ON ABCISSA AND S ON'/
+ 28X,' ORDINATE IN OUTPUT EDIT')
  WRITE(IOT3,307) IPL
307 FORMAT(/5X,'IPL =',I2,12X,' -1-NO PLOTS REQUESTED'/
* 25X,' 1-CALL FOR OUTPUT PLOTS'/
* 28X,'ON TAPE22')
  WRITE(IOT3,306) ITPE
306 FORMAT(/5X,'ITPE =',I2,12X,' -1-READ FNK FROM TAPE24 AND'/
+ 28X,'COMPLETE CALCULATION'/

```

```

+ 25X,' 0=CREATE FNK ON TAPE24 AND '/'
+ 28X,'COMPLETE CALCULATION'/
+ 25X,' 1=CREATE FNK IN TAPE24 AND STOP')
RETURN
END
SUBROUTINE ALP4(FN,GN,HN,W,FTC,P,LNR,LKR)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LX,JT,JX,
* IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
* KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SIO,ERN,ERK,DC,S0,DS,X0,DX
COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT6,MX(2),MY(2),EXMX
DIMENSION GN(LNR,LKR),FN(LNR,LKR),HN(LNR,LKR),W(LNR),FTC(LKR)
* ,P(LKR)
FN(IR,IR)=1.0D0
W(IR)=1.0D0
IF(IS.EQ.2) GO TO 40
FN1=1.0D0
X1=FTC(IR)/B1
DO 20 N=1,LN
NR=IR+N
FN1=X1*FN1
FN(NR,IR)=FN1
W(NR)=FN(NR,IR)
20 CONTINUE
GN(IR,IR+1)=1.0D0/3.0D0
GO TO 62
40 CONTINUE
CALL LEG(P,AM0,LK,LKR)
DO 59 LXX=1,LKP
L=LXX-1
LR=IR+L
FN(IR,LR)=P(LR)
59 CONTINUE
W(IR)=FN(IR,IR)
DO 60 N=1,LN
NR=IR+N
DO 61 LXX=1,LLP1
L=LXX-1
LR=IR+L
X1=FTC(LR)/B1
FN(NR,LR)=(X1**N)*P(LR)
61 CONTINUE
W(NR)=FN(NR,IR)
60 CONTINUE
41 CONTINUE
DO 71 LXX=1,LKP
L=LXX-1
LR=IR+L
GN(IR,LR)=AM0*P(LR)
71 CONTINUE
62 CONTINUE
DO 30 N=1,LN
NR=IR+N
DO 31 LXX=1,LKP
L=LXX-1

```

```

    LR=IR+L
    T1=(L+1.0D0)*FN(NR,IR+1)
    T2=L*FN(NR,LR-1)
    T3=N*FTC(LR)*GN(NR-1,LR)
    GN(NR,LR)=((T1+T2)/(2*L+1.0D0)+T3)/(N*B1+1.0D0)
31  CONTINUE
    W(NR)=GN(NR,IR)
30  CONTINUE
    WRITE(IOT4,*) (W(NX),NX=IR,LNR)
    DO 100 K=2,LK
    KR=IR+K
    AK2=2*K-1
    LU=LKP-K
    IF(IS.EQ.1) GO TO 52
    DO 51 LXX=1,LKP
    L=LXX-1
    LR=IR+L
    HN(IR,LR)=P(KR)*P(LR)
51  CONTINUE
    GO TO 53
52  CONTINUE
    DO 54 LXX=1,LKP
    L=LXX-1
    HN(IR,IR+L)=0.0D0
54  CONTINUE
    HN(IR,KR)=1.0D0/(AK2+2.0D0)
53  CONTINUE
    DO 101 N=1,LN
    NR=IR+N
    DO 102 LXX=1,LU
    L=LXX-1
    LR=IR+L
    L2=2*L+1
    T1=(N*B1+1.0D0-K)*FN(NR,LR)
    T2=L*GN(NR,LR-1)
    T3=(L+1.0D0)*GN(NR,LR+1)
    SS3=(T2+T3)*AK2/L2
    T4=N*FTC(LR)*(HN(NR-1,LR)-FN(NR-1,LR))
    SS5=T1+SS3+T4
    HN(NR,LR)=SS5/(K+N*B1)
102 CONTINUE
    W(NR)=HN(NR,IR)
101 CONTINUE
    WRITE(IOT4,*) (W(NX),NX=IR,LNR)
    DO 201 NX=1,LNP
    N=NX-1
    NR=IR+N
    DO 202 LXX=1,LU
    L=LXX-1
    LR=IR+L
    FN(NR,LR)=GN(NR,LR)
    GN(NR,LR)=HN(NR,LR)
202 CONTINUE
201 CONTINUE
100 CONTINUE
    RETURN

```

END

SUBROUTINE SCALR(XX,W,PHI,AMERR,PE,QU,FTC,U,LNR,LKR,LA,LO,LX,LLR)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C****

C NUMERICAL ALGORITHM

C CALCULATES FLUX DISTRIBUTION PHI

C FOR ALL OPTIONS

C PREPARES ARRAYS FOR OUTPUT EDIT

C****

```

COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LXX,JT,JX,
* IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
* KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SIO,ERN,ERK,DC,S0,DS,X0,DX
COMMON/C3/PI,X,S,Q,QY,TX, SX1,ET,TS,PH1,FD,DPH,WLP1
COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX
COMMON/C5/ PH0,ETS,ET1,A2T
DIMENSION W(LNR,LKR),PHI(LA,LO),AMERR(LA,LO),XX(LX),PE(LKR)
* ,QU(LKR),FTC(LLR),U(2,LKR)
DATA KFLG/1/
IF(KFLG.LT.0) GO TO 181
REWIND IOT4
DO 180 K=KFF,LK,KII
KX=IR+K
IF(KII.EQ.2) READ(IOT4,*) (W(NX,1),NX=IR,LNR)
READ(IOT4,*) (W(NX,KX),NX=IR,LNR)
180 CONTINUE
KFLG=-1
181 CONTINUE
ERNT=ERN
NKX=0
AMXN=1.0E-03
DO 1001 IX=1,LX
NDI=-DLOG10(ERK)
IMXX=1
SKL=1.0D10
X=XX(IX)
WRITE(IOT06,*) ' BEGIN X=',X
IA=MX(1)*IST+MX(2)*IX
IO=MY(1)*IST+MY(2)*IX
ET=X/S
ETS=1-ET*ET
ET1=1-ET
DET=DABS(ET)
IF(DABS(DET-1.0D0).LT.1.0E-12) GO TO 976
IF(DET.GT.1.0D0) GO TO 977
INF=1
CALL ANACOL(FTC,QU,PE,LKR,LLR)
IF(INF.LT.0) GO TO 979
AU=DEXP(QY*Q/B1)-1.0D0-FD
W(1,IR)=AU
U(1,IR)=AU
CALL LEG(PE,ET,LK,LKR)
155 CONTINUE
AERS=ERN
A2T=1.0D10
A2=A2T

```

```

ALT=A2T
IEE=0
SK=PH0+PH1+AU/2.0D0
IFLG1=1
IF(IMXX.LT.0) NKX=0
M=0
DO 140 K=KFF,LK,KII
KX=IR+K
IF(Q.EQ.0.0D0) GO TO 172
IF(K.LE.NKX) GO TO 152
A1=1.0D10
SN=0.0D0
IE=0
TZ=1.0D0
IF(NS.EQ.2) TZ=Q*QY
DO 150 N=NS,LN
TZ=Q*QY*TZ/N
TN=TZ*W(IR+N,KX)
SN=SN+TN

```

C***

C DIAGNOSTIC PRINT 1

```

C      WRITE(21,97) N,TN,SN,A1
      97 FORMAT(1X,I4,5(1PE12.4))

```

C***

```

      IF(SN.NE.0.0) A1=DABS(TN/SN)
      IF(A1.LT.ERN) IE=IE+1
      AERN=N*1.0E-3
      IF(IE.EQ.IEN) GO TO 151
150 CONTINUE
151 CONTINUE
      W(1,KX)=SN
152 CONTINUE
      IF(AERN.GT.AMXN) AMXN=AERN
      SR=(W(1,KX)-W(1,KX-2)*JTI)
      PKK=PE(KX-JTI)
      TK=(2*K*(1-JTI)+1.0D0)*SR*PKK/2.0D0
      W(2,KX)=TK
      U(1,KX)=SR
      SK=SK+TK
      W(3,KX)=SK
      IF(SK.NE.0.0D0) A2=DABS(TK/SK)
      IF(A2.LT.ERK.AND.ALT.LT.ERK) IEE=IEE+1
      ALT=A2
      IF(A2.GT.A2T.OR.K.LT.LK/4) GO TO 142
      NKT=K
      IF(A2.GT.1.0E-15) A2T=A2
142 CONTINUE

```

C***

C DIAGNOSTIC PRINT 2

```

C      WRITE(21,97) K,TK,SK,A2

```

C***

```

      NKX=K
      IF(IEE.EQ.IEK) GO TO 141
140 CONTINUE
      IF(SK.EQ.0.0) GO TO 172
      IFLG1=-1

```

```

      SK=W(3,IR+NKT)
141  CONTINUE
      T1L=DABS(W(2,IR+KFF))
      DO 161 K=KFF,NKX,KII
      T1=DABS(W(2,IR+K))
      IF(T1.GT.T1L) T1L=T1
161  CONTINUE
      T1L=T1L/DABS(SK)
      IF(T1L.LE.1.0D0.OR.T1L.EQ.0.0) GO TO 162
      AT1=DLOG10(T1L)
      E1=DABS((SK-SKL)/SK)
      IF(E1.LT.ERK) GO TO 162
      SKL=SK
      IMXX=-1
      LM1=AT1+1+NDI
      LM2=-DLOG10(ERN)
      IF(LM2.GT.LM1.OR.LM1.GT.20) GO TO 162
      ERN=1.0D0/10.0D0**LM1
      NDI=NDI+1
      GO TO 155
162  CONTINUE
      IF(IFLG1.GT.0.OR.LM.EQ.0) GO TO 163
      CALL ACC(U,W,PE,SK,LNR,LKR,M)
163  CONTINUE
      ERN=ERNT
      IF(IFLG1.GT.0) GO TO 172
      X1=-DLOG10(A2T)+1
      IX1=X1
      AERS=10.0**IX1
172  CONTINUE
      IF(IFLG1.LT.0) NKX=NKT
      IF(NKX.LT.10) NKX=10
      AMERR(IA,IO)=(NKX+AMXN+M*1.0E-04)*IMXX*AERS
      PHI(IA,IO)=TX*SK*SX1*Q0
      SKP=PHI(IA,IO)
      IF(IDFF.GT.0) GO TO 1001
      CALL DIFF
      PHI(IA,IO)=DABS((SKP-DPH)/SKP)
135  CONTINUE
      GO TO 1001
976  CONTINUE
      IF(JTI.EQ.1) GO TO 87
      PHI(IA,IO)=TX*Q0/2.0D0
      IF(IS.EQ.1) GO TO 978
      PHI(IA,IO)=0.0D0
      IF(DABS(ET*AM0-1.0D0).LT.1.0E-12) GO TO 979
      GO TO 978
87   CONTINUE
      PHI(IA,IO)=0.0D0
      IF(IS.EQ.2.AND.AM0.EQ.1.0D0) PHI(IA,IO)=Q0*TS
      IF(ET.LT.0.0) PHI(IA,IO)=Q0
978  CONTINUE
      IF(IDFF.GT.0) GO TO 1001
      IF(PHI(IA,IO).EQ.0.0) GO TO 1001
      CALL DIFF
      PHI(IA,IO)=DABS((PHI(IA,IO)-DPH)/PHI(IA,IO))

```

```

      GO TO 1001
977  CONTINUE
      IF(ET.GT.1.0.OR.JTI.EQ.0) GO TO 1001
      PHI(IA,IO)=Q0
      GO TO 978
979  CONTINUE
      PHI(IA,IO)=7.7777E17
      IF(IDFF.GT.0) GO TO 1001
      PHI(IA,IO)=1.0D0
1001 CONTINUE
      RETURN
      END
      SUBROUTINE ANACOL(FTC,QU,PE,LKR,LLR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LXX,JT,JX,
      * IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
      * KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
      COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SIO,ERN,ERK,DC,S0,DS,X0,DX
      COMMON/C3/PI,X,S,Q,QY,TX, SX1,ET,TS,PH1,FD,DPH,WLP1
      COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX
      COMMON/C5/ PH0,ETS,ET1,A2T
      DIMENSION FTC(LLR),QU(LLR),PE(LKR)
      IF(JTI.EQ.1) GO TO 10
      IF(IS.EQ.1) GO TO 11
      PH0=0.0D0
      IF(ET.NE.AM0) GO TO 200
      INF=-1
      GO TO 200
11  CONTINUE
      PH0=0.5D0
      GO TO 200
10  CONTINUE
      IF(IS.EQ.1) GO TO 15
      PH0=0.0D0
      IF(AM0.GE.ET) PH0=1.0D0
      GO TO 200
15  CONTINUE
      PH0=ET1/2.0D0
200 CONTINUE
      PH1=0.0D0
      FD=0.0D0
      IF(NS.NE.2) RETURN
      CALL FCOLA(FTC,QU,PE,LKR,LLR)
      FD=Q
      RETURN
      END
      SUBROUTINE FCOLA(FTC,QU,PE,LKR,LLR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*****
: FIRST COLLIDED FLUX FOR IS=2
: DETERMINES FIRST COLLIDED FLUX
: FOR BEAM SOURCE
*****
      COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LX,JT,JX,
      * IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
      * KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT

```

```

COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,S10,ERN,ERK,DC,S0,DS,X0,DX
COMMON/C3/PI,X,S,Q,QY,TX,SX1,ET,TS,PH1,FD,DPH,WLP1
COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX
DIMENSION FTC(LLR),QU(LLR),PE(LKR)
CALL LEG(PE,AM0,LL,LKR)
DO 2 KP=1,LLP1
  K=KP-1
  QU(IR+K)=PE(IR+K)
2 CONTINUE
  CALL LEG(PE,ET,LL,LKR)
  IMF=1
  IF(AM0.LT.ET) IMF=0
  IMS=1
  IF(AM0.GT.ET) IMS=0
  D1=DABS(ET-AM0)
  IF(D1.LT.1.0E-12) GO TO 101
  D2=1.0D0-AM0
  T1=0.0D0
  T2=0.0D0
  IF(D2.GT.0.0D0) T1=DLOG(D2/D1)
  D3=1.0D0+AM0
  IF(D3.GT.0.0D0) T2=DLOG(D3/D1)
  P0=IMF*T2+IMS*T1
  P1=AM0*P0-(1.0D0+ET)*IMF+(1.0D0-ET)*IMS
  SL=FTC(IR)*P0/2.0D0
  IF(LL.EQ.0) GO TO 100
  SL=SL+3.0D0*FTC(IR+1)*AM0*P1/2.0D0
  IF(LL.EQ.1) GO TO 100
  DO 1 L=2,LL
    T1=(2*L-1)*AM0*P1/L
    T2=(L-1)*P0/L
    T3=(PE(IR+L)-PE(IR+L-2))/L
    P2=T1-T2-T3
    P0=P1
    P1=P2
  SL=SL+(2*L+1)*FTC(IR+L)*QU(IR+L)*P2/2.0D0
1 CONTINUE
100 CONTINUE
  PH1=Q*SL
  RETURN
101 CONTINUE
  INF=-1
  RETURN
END
SUBROUTINE ACC(U,W,PE,SK,LNR,LKR,MU)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LXX,JT,JX,
* IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
* KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
  COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,S10,ERN,ERK,DC,S0,DS,X0,DX
  COMMON/C3/PI,X,S,Q,QY,TX,SX1,ET,TS,PH1,FD,DPH,WLP1
  COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX
  COMMON/C5/ PH0,ETS,ET1,A2T
  DIMENSION W(LNR,LKR),U(2,LKR),PE(LKR)
  A2ML=1.0D10
  NKTL=NKT

```

```

IFLG1=1
A2T=1.0D10
ALT=A2T
LKMM=LK
IEE=0
SKL=SK
DO 500 M=1,LM
MU=M
X1=1.0D0/(2.0D0*(ETS**M))
U(2,IR)=2.0D0*(U(1,IR)-U(1,IR+2))/3.0D0
SK= U(2,IR)*X1+PH0+PH1
W(3,IR)=SK
LKMM=LKMM-2
DO 501 K=KFF,LKMM,KII
KX=IR+K
G1=(K+1)*(K+2.0D0)/((2*K+1)*(2*K+3))
G2=(K+1)*(K+1.0D0)/((2*K+1)*(2*K+3))
G2=G2+K*K/((2*K-1.0D0)*(2*K+1))
G3=(K-1)*K/((2*K-1.0D0)*(2*K+1))
U(2,KX)=U(1,KX)*(1-G2)-G1*U(1,KX+2)-G3*U(1,KX-2)
TK=U(2,KX)*(2*K+1)*PE(KX)*X1
SK=SK+TK
W(3,KX)=SK
A2=DABS(TK/SK)
IF(A2.LT.ERK.AND.ALT.LT.ERK) IEE=IEE+1
ALT=A2
IF(A2.GT.A2T.OR.K.LT.LK/4) GO TO 542
NKT=K
IF(A2.GT.1.0E-15) A2T=A2
542 CONTINUE
C***
C DIAGNOSTIC PRINT 3
C WRITE(21,988) M,K,TK,SK,A2
988 FORMAT(2I4,3(1PE12.5))
C***
NKX=K
IF(IEE.EQ.IEK) GO TO 541
501 CONTINUE
U(1,IR)=U(2,IR)
DO 502 K=KFF,LKMM,KII
KX=IR+K
U(1,KX)=U(2,KX)
502 CONTINUE
SK=W(3,IR+NKT)
IF(A2T.GT.A2ML) GO TO 500
SKL=SK
A2ML=A2T
NRTL=NKT
500 CONTINUE
IFLG1=-1
SK=SKL
NKT=NRTL
A2T=A2ML
541 CONTINUE
RETURN
END

```

```

SUBROUTINE SRUTH(FTC,LMR)
C SCREENED RUTHERFORD LEGENDRE SCATTERING COEFFICIENTS
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C****
COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LYY,JT,JX,
* IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
* KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SIO,ERN,ERK,DC,S0,DS,X0,DX
COMMON/C3/PI,X,S,Q,QY,TX, SX1,ET,TS,PH1,FD,DPH,WLP1
COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX
DIMENSION FTC(LMR)
FTC(IR)=1.0D0
IF(LL.EQ.0) RETURN
FTC(IR+1)=1+ETA-ETA*(1.0D0+ETA/2.0D0)*DLOG(1.0D0+2.0D0/ETA)
IF(LL.EQ.1) RETURN
AX=FTC(IR+1)
FC=DEXP(-EXMX)
RT=1.0D0+ETA-DSQRT(ETA*(ETA+2.0D0))
ALP=0.25D0*(5.0D0+RT*RT)/(1.0D0-RT*RT)
L1=MLT+LLP2
AL1=L1
W0=FC*RT*DSQRT(AL1/ALP+1.0D0)
W1=FC*DSQRT((AL1-1.0D0)/ALP+1.0D0)
LMX=-EXMX/DLOG(RT)
LSG=LL
IF(LMX.GT.L1) GO TO 144
L1=LMX
LSG=L1-MLT-2
WRITE(IOT1,88) LSG
88 FORMAT('LEGENDRE SERIES TRUNCATED AT LL=',I4)
144 CONTINUE
DO 50 LP=1,L1
  L=L1-LP+1
  W2=((2*L+3)*(1+ETA)*W1-(L+1)*W0)/(L+2)
  FTC(IR+L)=W2
  W0=W1
  W1=W2
50 CONTINUE
GX=AX/FTC(IR+1)
FTC(IR+1)=AX
DO 51 L=2,LSG
  FTC(IR+L)=GX*FTC(IR+L)
51 CONTINUE
RETURN
END
SUBROUTINE TRNSC(FTC,LLR)
TRANSPORT CORRECTED LEGENDRE
SCATTERING COEFFICIENTS
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LXX,JT,JX,
* IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
* KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SIO,ERN,ERK,DC,S0,DS,X0,DX
COMMON/C3/PI,X,S,Q,QY,TX, SX1,ET,TS,PH1,FD,DPH,WLP1
COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT06,MX(2),MY(2),EXMX
COMMON/C5/ PH0,ETS,ET1,A2T

```

```

        DIMENSION FTC(LLR)
        WLP1=FTC(LLR)
        T1=1.0D0-WLP1
        IF(LL.LE.1) GO TO 11
        DO 10 L=1,LL
        FTC(IR+L)=(FTC(IR+L)-WLP1)/T1
10      CONTINUE
11      CONTINUE
        ALA=ALA/T1
        LL=LL-1
        LLP1=LL+1
        LLP2=LL+2
        LLR=LLR-1
        RETURN
        END
        SUBROUTINE DIFF
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C****
C  DIFFUSION CALCULATION
C  DIFFUSION COEFFICIENT=DC
C****
        COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LX,JT,JX,
        * IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
        * KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
        COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SI0,ERN,ERK,DC,S0,DS,X0,DX
        COMMON/C3/PI,X,S,Q,QY,TX, SX1,ET,TS,PH1,FD,DPH,WLP1
        COMMON/C5/ PH0,ETS,ET1,A2T
        IF(JTI.GT.0) GO TO 10
        T1=X*X/(4.0D0*DC*S)
        T2=DEXP(-T1)
        T3=4.0D0*PI*DC*S
        T3=T3**0.5
        DPH=Q0*T2/T3
        IF(IS.EQ.1) RETURN
        DPH=DPH*(1+3.0D0*AM0*X/(2.0D0*S))
        RETURN
10      CONTINUE
        X1=X/DSQRT(4.0D0*DC*S)
        CALL ERF(X1,ER)
        DPH=Q0*(1.0D0-ER)/2.0D0
        IF(IS.EQ.1) RETURN
        T0=3.0D0*AM0*DSQRT(DC/(PI*S))
        DPH=DPH+Q0*T0*DEXP(-X1*X1)/2.0D0
        RETURN
        END
        SUBROUTINE LEG(P,AX,LK,LKR)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
****
:  LEGENDRE POLYNOMIAL ROUTINE
:  DETERMINES LEGENDRE POLYNOMIALS
:  FOR L=0,LK
:****
        COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LX,JT,JX,
        * IDFF,LN,LNP,LKX,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
        * KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
        DIMENSION P(LKR)

```

```

P(IR)=1.0D0
P(IR+1)=AX
IF(LK.LE.1) RETURN
DO 10 J=2,LK
JR=IR+J
PXX=((2*J-1.0D0)*AX*P(JR-1)-(J-1.0D0)*P(JR-2))/J
P(JR)=PXX
10 CONTINUE
RETURN
END
SUBROUTINE ERF(X,ERFF)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION A(5)
DATA ITE/-1/
IF(ITE.GT.0) GO TO 10
A(1)=0.254829592
A(2)=-0.284496736
A(3)=1.421413741
A(4)=-1.453152027
A(5)=1.061405429
P=0.3275911
ITE=1
10 CONTINUE
T=1.0D0/(1.0D0+P*X)
ERFF=0
DO 1 M=1,5
ERFF=ERFF+A(M)*(T**M)
1 CONTINUE
ERFF=1.0D0-ERFF*DEXP(-X*X)
RETURN
END
SUBROUTINE OUT(AB,OR,FTC,PHI,AMERR,LA,LO,LLR)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

C****

C OUTPUT EDIT ROUTINE

```

COMMON/C1/IR,IS,JTI,LL,ITRN,LLP1,LLP2,ITC,IPM,LS,LX,JT,JX,
* IDFF,LN,LNP,LK,LKP,IEN,IEK,ITR,IP,IXY,IPL,ITPE,NS,
* KFF,KII,LM,IST,IN,INF,NKT,IFLG1,NKX,IMXM,LAP,LOP,MLT
COMMON/C2/IDEN(18),Q0,AM0,ETA,ALA,BE,B1,SI0,ERN,ERK,DC,S0,DS,X0,DX
COMMON/C3/PI,X,S,Q,QY,TX,SX1,ET,TS,PH1,FD,DPH,WLP1
COMMON/C4/IOT1,IOT2,IOT3,IOT4,IOT6,MX(2),MY(2),EXMX
DIMENSION PHI(LA,LO),AMERR(LA,LO),AB(LA),OR(LO)
* ,FTC(LLR)
REAL FS(3),FL(5),YC,XC
DATA ITU/1/
WRITE(IOT1,399) IMXM
399 FORMAT(/1X,'MINIMUM CONTAINER ARRAY DIMENSION=',I6)
IF(IPL.NE.1) GO TO 400
PHMX=0.0D0
DO 401 I=1,LAP
DO 402 J=1,LOP
IF(PHI(I,J).EQ.7.7777E17) GO TO 402
IF(DABS(PHI(I,J)).GT.PHMX) PHMX=DABS(PHI(I,J))
402 CONTINUE
401 CONTINUE

```

```

400 CONTINUE
    IF(JTI.EQ.1) GO TO 16
    FL(1)=4HAT X
    FL(2)=2H=0
    IJ=2
    GO TO 14
16 CONTINUE
    FL(1)=4HCONS
    FL(2)=4HTANT
    FL(3)=4H HAL
    FL(4)=4HF-SP
    FL(5)=3HACE
    IJ=5
14 CONTINUE
    IF(IS.EQ.2) GO TO 18
    FS(1)=4HISOT
    FS(2)=4HROPI
    FS(3)=1HC
    GO TO 15
18 CONTINUE
    FS(1)=4HANIS
    FS(2)=4HOTRO
    FS(3)=3HPIC
15 CONTINUE
    IF(MX(1).EQ.1) XC=1HS
    IF(MX(2).EQ.1) XC=1HX
    IF(MY(1).EQ.1) YC=4H S/
    IF(MY(2).EQ.1) YC=4H X/
17 CONTINUE
    NT=LAP/6
    LTX=5
    IUX=6
    IF(ITR.GT.0) GO TO 36
    NT=LAP/11
    LTX=10
    IUX=11
36 CONTINUE
    NX=LAP-IUX*NT
    IF(NX.GT.0) NT=NT+1
    IF(ITU.LT.0) GO TO 333
    WRITE(IOT1,222)
222 FORMAT(/1X,'PLANE GEOMETRY '//1X,'SOURCE:')
    WRITE(IOT1,23) (FL(I),I=1,IJ)
23 FORMAT(3X,6A)
    WRITE(IOT1,230) (FS(I),I=1,3)
230 FORMAT(3X,6A)
    IF(IS.EQ.2) WRITE(IOT1,38) AM0
38 FORMAT(3X,'MU0=',1PE12.4)
    WRITE(IOT1,66) LL,(FTC(LY),LY=IR,LLR)
66 FORMAT(/1X,'SCATTERING ORDER LL=',I4/
* ' LEGENDRE COEFFICIENTS:'/
* (1X,5(1PE12.5)))
    IF(ITC.GT.0) GO TO 84
    WRITE(IOT1,85) ETA,ALA
85 FORMAT(/1X,'ETA=',1PE12.4,2X,'ALA=',1PE12.4)
    GO TO 86

```

```

84 CONTINUE
  WRITE(IOT1,87) ALA
87 FORMAT(/1X,'ALA=',1PE12.4)
86 CONTINUE
  WRITE(IOT1,81) BE,SIO
81 FORMAT(/1X,' BE =',1PE12.5,2X,'SIO=',1PE12.5)
  IF(IDFF.LT.0) WRITE(IOT1,65)
65 FORMAT(/1X,'DIFFUSION THEORY COMPARISON')
  IF(IDFF.GT.0) WRITE(IOT1,74)
74 FORMAT(/1X,'FLUX CALCULATION')
333 CONTINUE
  DO 1 LLT=1,NT
    IF(ITR.LT.0)
      +WRITE(IOT1,32)
32 FORMAT(/)
    IF(IPL.EQ.1) WRITE(IOT2,34) LLT,LAP,LOP
34 FORMAT(1X,'FILE',I2,2I4)
    IX1=1+IUX*(LLT-1)
    IF(NX.EQ.0) GO TO 35
    IF(LLT.EQ.NT) LTX=NX-1
35 CONTINUE
    IX2=LTX+IX1
    WRITE(IOT1,20) YC,XC, (AB(IX),IX=IX1,IX2)
    IF(IPL.EQ.1)
      *WRITE(IOT2,200) YC,XC, (AB(IX),IX=IX1,IX2)
C****
C FORMAT MAY BE CHANGED TO ACCOMMODATE PLOTTING PACKAGE
200 FORMAT(2X,2A4,1X,6(1PE11.4))
~****
20 FORMAT(/2X,A,A,1X,10(1PE11.4))
  IF(ITR.GT.0)
    +WRITE(IOT1,21)
21 FORMAT(1X,38(2H**),1H*)
  IF(ITR.LT.0)
    +WRITE(IOT1,22)
22 FORMAT(1X,61(2H**))
  DO 30 IU=1,LOP
    WRITE(IOT1,31) OR(IU),(PHI(IY,IU),IY=IX1,IX2)
31 FORMAT(1PE11.4,11(1PE11.4))
  IF(IPL.NE.1) GO TO 30
  DO 300 ICK=IX1,IX2
    IF(PHI(ICK,IU).EQ.7.7777E17) PHI(ICK,IU)=5.0D0*PHMX
300 CONTINUE
    WRITE(IOT2,331) OR(IU),(PHI(IY,IU),IY=IX1,IX2)
]****
] FORMAT MAY BE CHANGED TO ACCOMMODATE PLOTTING PACKAGE
331 FORMAT(1PE11.4,6(1PE11.4))
****
30 CONTINUE
  IF(ITR.GT.0)
    +WRITE(IOT1,21)
  IF(ITR.LT.0)
    +WRITE(IOT1,22)
  WRITE(IOT1,37)
37 FORMAT(//6X,'NUMBER OF TERMS FOR CONVERGENCE')
  DO 40 IU=1,LOP

```

May 30 11:35 1990 sleetm.for Page 23

```
      WRITE(IOT1,33) (AMERR(IY,IU),IY=IX1,IX2)
33  FORMAT(6X,11(2PE12.5))
40  CONTINUE
1   CONTINUE
    ITU=-1
    RETURN
    END
```

3 The F_N Method Applied to Neutral Particle Transport in Multiple Slabs: A Manual

Abstract

I. Introduction

II. Coupled Integral Equations for the Exiting Flux

III. The F_N Approximation

A. Single Slab

B. Multiple Slabs

C. Exponential Source

IV. Numerical Implementation and Demonstration

A. Numerical Considerations

B. Demonstration

V. Code Operation

A. Flow Chart

B. Input Description

1. Sample Problem Input

C. Output Description

1. Sample Problem Output

Sample Problem 1

Sample Problem 2

Sample Problem 3

D. Detailed Program Notes

1. Main Program

1. FNX Subprogram

Acknowledgement

References

The F_N Method Applied to Neutral Particle Transport in Multiple Slabs: A Manual

B. D. Ganapol¹

Department of Nuclear and Energy Engineering
University of Arizona

Abstract

This document describes the operation of the MSLAB program which embodies the F_N algorithm for 1-D neutral particle transport in heterogeneous slab geometry. The F_N algorithm was first developed by C. E. Siewert and has been shown to be one of the most robust of any of the 1-D transport methods. The theory and numerical development of the algorithm as well as programming notes and sample problems are included.

I Introduction

The determination of the dose distribution resulting from electron energy deposition along the path of travel in some applications requires the solution of the neutral particle transport equation. Currently, there are several numerical solution methodologies that provide reliable solutions to the electron transport equation including Monte Carlo, eigenfunction expansions, P_N theory, numerical transform inversion, and S_N theory. The last of these, discrete ordinates or S_N theory, has proven to be a particularly valuable approach since it is more cost effective than most others. However, even with its efficiency and apparent accuracy, the S_N method requires complete discretization of the independent variables (space and direction) leading to the question of just how reliable is the S_N solution. For this reason, it is imperative that highly accurate independent solutions continue to be developed in order to assess the numerical discretization error inherent in the discrete ordinate approximation.

¹Work performed for the CUBUC contract (No S-0-7542) for Rome Air Development Center, Hanscom Air Force Base.

This manual gives the details of one such numerical monoenergetic, 1-D transport determination, the F_N method, which was originally developed by C. E. Siewert [1] but was never released as an operational code. The major advantage of the F_N method is that only a single approximation (in the angular variable) need be made allowing the independent variables to be treated as continuous.

This manual will delineate in detail the mathematical derivation, numerical analysis, programming, and the usage of the resultant program. Section II provides the mathematical analysis required to generate the coupled equations from the transport equation that lead to the boundary fluxes. In Section III the F_N approximation for a single slab and multiple slabs with isotropic sources is given. Section IV details the numerical analysis required to implement the F_N method in FORTRAN 77. The last section details the operation of the code MSLAB including a flow chart, explanation of the input deck plus several examples, explanation of the output generated, and output from the examples. In addition, detailed programming notes deemed necessary by the author are provided.

II Coupled Integral Equations for the Exiting Flux

With only minor differences, the theory associated with the F_N method applied here, is contained in [1]. For clarity, the major points of the theory will be summarized.

The one-group neutral particle transport equation to be solved in a slab of width Δ for $|\mu| \leq 1$ and $a \leq x \leq b$ is

$$\left[\mu \frac{\partial}{\partial \tau} + 1 \right] \Phi(\tau, \mu) = \frac{c}{2} \int_{-1}^{+1} d\mu' \Phi(\tau, \mu') + \frac{1}{2} S(\tau) \quad (1a)$$

where isotropic scattering and an isotropically emitting spatial source have been assumed. The boundary conditions are

$$\Phi(a, \mu) = F_L(\mu), \quad \mu > 0 \quad (1b)$$

$$\Phi(b, \mu) = F_R(-\mu), \quad \mu < 0 \quad (1c)$$

with F_L and F_R general functions of μ . These functions are assumed known but they may actually depend upon the transmitted and reflected fluxes from adjacent slabs as indicated

below. Indeed, it will be the boundary fluxes that connect all slabs together as shown in Section III.B.

As a starting point, the r_N method requires integral equations for the reflected and transmitted fluxes from the slab. These equations are derived by manipulation of eqs. (1) as follows:

1. Let μ be replaced by $-\mu$, multiply by $e^{-\tau/s}$ and integrate over τ on $[a, b]$ to give

$$\frac{s\mu}{\mu-s}B(\mu, s) - \int_a^b d\tau \Phi(\tau, -\mu)e^{-\tau/s} = \frac{c}{2} \frac{s}{\mu-s} \rho^*(s) + \frac{1}{2} \frac{s}{\mu-s} S^*(s, a, b) \quad (2a)$$

where,

$$B(\mu, s) \equiv \Phi(a, -\mu)e^{-a/s} - \Phi(b, -\mu)e^{-b/s} \quad (2b)$$

$$\rho^*(s) \equiv \int_a^b d\tau e^{-\tau/s} \int_{-1}^{+1} d\mu \Phi(\tau, -\mu) \quad (2c)$$

$$S^*(s, a, b) = \int_a^b d\tau S(\tau)e^{-\tau/s} \quad (2d)$$

and s is a complex variable in the cut plane $[-1, 1]$.

2. Integrate eq. (2a) over μ on $[-1, 1]$ to obtain

$$\int_{-1}^{+1} d\mu \frac{\mu}{\mu-s} B(\mu, s) = \frac{\Lambda(s)\rho^*(s)}{s} + L(s)S^*(s, a, b) \quad (3a)$$

where,

$$L(s) \equiv \frac{1}{2} \int_{-1}^{+1} d\mu \frac{1}{\mu-s} = \frac{1}{2} \ln \left(\frac{s-1}{s+1} \right) \quad (3b)$$

$$\Lambda(s) \equiv 1 + \frac{cs}{2} \int_{-1}^{+1} d\mu \frac{1}{\mu-s} = 1 + \frac{cs}{2} \ln \left(\frac{s-1}{s+1} \right). \quad (3c)$$

3. Multiply eq. (3a) by $e^{a/s}$ and restrict s to $\text{Re}(s) > 0$ to give

$$\int_{-1}^{+1} d\mu \frac{\mu}{\mu-s} C(\mu, s) = \Lambda(s) \frac{I^*(s)}{s} + L(s)S^*(s, a, b)e^{a/s} \quad (4a)$$

where

$$I^*(s) \equiv \int_a^b d\tau e^{-\frac{\tau-a}{s}} \int_{-1}^{+1} d\mu \Phi(\tau, -\mu) \quad (4b)$$

$$C(\mu, s) \equiv \Phi(a, -\mu) - \Phi(b, -\mu)e^{-\frac{b-a}{s}}. \quad (4c)$$

4. To obtain a second equation, replace s by $-s$ and μ by $-\mu$ in eq. (3a) and multiply it by $e^{-b/s}$, again with $\text{Re}(s) > 0$ to give

$$\int_{-1}^{+1} d\mu \frac{\mu}{\mu - s} D(\mu, s) = \Lambda(s) \frac{J^*(s)}{s} + L(s) S^*(-s, a, b) e^{-\frac{b}{s}} \quad (5a)$$

where

$$J^*(s) \equiv \int_a^b d\tau e^{-\frac{b-\tau}{s}} \int_{-1}^{+1} d\mu \Phi(\tau, -\mu) \quad (5b)$$

$$D(\mu, s) \equiv \Phi(b, \mu) - \Phi(a, \mu) e^{-\frac{a}{s}}. \quad (5c)$$

5. In addition, since

$$\Lambda(\pm\nu_0) = 0, \quad \nu_0 \geq 1 \quad (6)$$

when $s = \nu_0$ in eq. (4a) and eq. (5a), we have

$$-\frac{c\nu_0}{2} \int_{-1}^{+1} d\mu \frac{\mu}{\mu - \nu_0} C(\mu, \nu_0) = \frac{1}{2} S^*(\nu_0, a, b) e^{-\frac{a}{\nu_0}} \quad (7a)$$

$$-\frac{c\nu_0}{2} \int_{-1}^{+1} d\mu \frac{\mu}{\mu - \nu_0} D(\mu, \nu_0) = \frac{1}{2} S^*(-\nu_0, a, b) e^{-\frac{b}{\nu_0}} \quad (7b)$$

Equations (4a) and (5a) are coupled integral equations with constraints given by eqs. (7). These integral equations will now be restricted to s on the cut $[0, 1]$. This is accomplished using the Plemelj relations in the form

$$\lim_{\epsilon \rightarrow 0} \left(\frac{1}{\mu - (\nu \pm i\epsilon)} \right) = \mathcal{P} \left[\frac{1}{\mu - \nu} \right] \pm i\pi \delta(\mu - \nu)$$

where $\nu, \epsilon, [0, 1]$ and $\mathcal{P} []$ represents a principal value when under an integral. Thus, if $\chi(s)$ is defined as the integral

$$\chi(s) = \int_{-1}^{+1} d\mu \frac{f(\mu)}{\mu - s}$$

then

$$\lim_{\epsilon \rightarrow 0} [\chi(\nu \pm i\epsilon)] \equiv \chi^{\pm}(\nu) = \mathcal{P} \int_{-1}^{+1} d\mu \frac{f(\mu)}{\mu - \nu} \pm i\pi f(\nu). \quad (8)$$

After application of eq. (8) to eq. (4a), we find

$$\mathcal{P} \int_{-1}^{+1} d\mu \frac{\mu}{\mu - \nu} C(\mu, \nu) \pm i\pi \nu C(\nu, \nu) = \frac{\Lambda^{\pm}(\nu) I^*(\nu)}{\nu} + L^{\pm}(\nu) S^*(\nu, a, b) e^{\frac{a}{\nu}} \quad (9a)$$

noting that I^* and S^* are analytic functions and

$$\Lambda^{\pm}(\nu) = \lambda(\nu) \pm i\pi \frac{c\nu}{2} \quad (9b)$$

$$\lambda(\nu) \equiv 1 + \frac{c\nu}{2} \ln \left| \frac{1+\nu}{1-\nu} \right| \quad (9c)$$

$$L^{\pm}(\nu) = L(\nu) \pm \frac{i\pi}{2} \quad (9d)$$

$$L(\nu) = \frac{1}{2} \ln \left| \frac{1-\nu}{1+\nu} \right|. \quad (9e)$$

Next, the plus and minus branches from eq. (9a) are subtracted to give

$$\frac{I^*(\nu)}{\nu} = \frac{2}{c} C(\nu, \nu) - \frac{e^{\frac{a}{\nu}}}{c\nu} S^*(\nu, a, b); \quad (10a)$$

the plus and minus branches of eq. (9a) are then added to give

$$\mathcal{P} \int_{-1}^{+1} d\mu \frac{\mu}{\mu - \nu} C(\mu, \nu) = \lambda(\nu) \frac{I^*(\nu)}{\nu} + L(\nu) S^*(\nu, a, b) e^{\frac{a}{\nu}}. \quad (10b)$$

Substitution of eq. (10a) into eq. (10b) then yields the first of the desired integral equations

$$\nu \lambda(\nu) C(\nu, \nu) = \frac{c\nu}{2} \mathcal{P} \int_{-1}^{+1} d\mu \frac{\mu}{\mu - \nu} C(\mu, \nu) + \frac{e^{\frac{a}{\nu}}}{2} S^*(\nu, a, b). \quad (11a)$$

Similarly, for eq. (5a), we obtain the second integral equation

$$\nu \lambda(\nu) D(\nu, \nu) = \frac{c\nu}{2} \mathcal{P} \int_{-1}^{+1} d\mu \frac{\mu}{\mu - \nu} D(\mu, \nu) + \frac{e^{-\frac{b}{\nu}}}{2} S^*(-\nu, a, b). \quad (11b)$$

Equations (11) are the integral equations to be solved subject to the constraints given by eqs. (7). The F_N approximation is now constructed.

III The F_N Approximation

III.A Single Slab

Equations (11) can be written in terms of the reflected and transmitted fluxes as follows:

$$\begin{aligned} \lambda(\nu) \Phi(a, -\nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} \Phi(a, -\mu) + \\ + \frac{c}{2} e^{-\frac{a}{\nu}} \int_0^1 d\mu \frac{\mu}{\mu + \nu} \Phi(b, \mu) = \frac{c}{2} \int_0^1 d\mu \frac{\mu}{\mu + \nu} F_L(\mu) + \\ + e^{-\frac{a}{\nu}} \left\{ \lambda(\nu) F_R(\nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} F_R(\mu) \right\} + \frac{e^{\frac{a}{\nu}}}{2\nu} S^*(\nu, a, b) \end{aligned} \quad (12a)$$

$$\begin{aligned} \lambda(\nu) \Phi(b, \nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} \Phi(b, \mu) + \\ + \frac{c}{2} e^{-\frac{b}{\nu}} \int_0^1 d\mu \frac{\mu}{\mu + \nu} \Phi(a, -\mu) = \frac{c}{2} \int_0^1 d\mu \frac{\mu}{\mu + \nu} F_R(\mu) + \\ + e^{-\frac{b}{\nu}} \left\{ \lambda(\nu) F_L(\nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} F_L(\mu) \right\} + \frac{e^{-\frac{b}{\nu}}}{2\nu} S^*(-\nu, a, b). \end{aligned} \quad (12b)$$

The corresponding equations for the constraints, eqs. (7), are obtained from equations (12) by letting ν be ν_0 and $\lambda(\nu_0)$ be zero.

The F_N approximation is simply

$$\Phi(a, -\mu) = F_R(\mu)e^{-\frac{\Delta}{\mu}} + \frac{c}{2} \sum_{\alpha=0}^{N-1} a_{\alpha} \psi_{\alpha}(\mu) \quad (13a)$$

$$\Phi(b, \mu) = F_L(\mu)e^{-\frac{\Delta}{\mu}} + \frac{c}{2} \sum_{\alpha=0}^{N-1} b_{\alpha} \psi_{\alpha}(\mu) \quad (13b)$$

where a_{α} , and b_{α} are coupling coefficients which are determined via collocation as will be shown; and $\psi_{\alpha}(\mu)$ are a set of known basis functions. When the F_N approximation is introduced into eqs. (12), we obtain

$$\sum_{\alpha=0}^{N-1} \left[a_{\alpha} B_{\alpha}(\nu) + b_{\alpha} A_{\alpha}(\nu) e^{-\frac{\Delta}{\nu}} \right] = R_1(\nu, a, b, \Delta) \quad (14a)$$

$$\sum_{\alpha=0}^{N-1} \left[a_{\alpha} A_{\alpha}(\nu) e^{-\frac{\Delta}{\nu}} + b_{\alpha} B_{\alpha}(\nu) \right] = R_2(\nu, a, b, \Delta) \quad (14b)$$

where

$$A_{\alpha}(\nu) \equiv \frac{c}{2} \int_0^1 d\mu \frac{\mu}{\mu + \nu} \psi_{\alpha}(\mu) \quad (14c)$$

$$B_{\alpha}(\nu) \equiv \lambda(\nu) \psi_{\alpha}(\nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} \psi_{\alpha}(\mu) \quad (14d)$$

$$R_1(\nu, a, b, \Delta) \equiv \int_0^1 d\mu \mu [S(\Delta, \mu, \nu) F_L(\mu) + C(\Delta, \mu, \nu) F_R(\mu)] + \frac{c}{c\nu} S^*(\nu, a, b) \quad (14e)$$

$$R_2(\nu, a, b, \Delta) \equiv \int_0^1 d\mu \mu [S(\Delta, \mu, \nu) F_R(\mu) + C(\Delta, \mu, \nu) F_L(\mu)] + \frac{c}{c\nu} S^*(-\nu, a, b) \quad (14f)$$

$$C(\Delta, \mu, \nu) \equiv \frac{e^{-\frac{\Delta}{\mu}} - e^{-\frac{\Delta}{\nu}}}{\mu - \nu} \quad (14g)$$

$$S(\Delta, \mu, \nu) \equiv \frac{1 - e^{-\Delta(\frac{1}{\nu} + \frac{1}{\mu})}}{\mu + \nu}. \quad (14h)$$

The coefficients a_α and b_α are obtained using collocation where ν is specified to be ν_0 plus the $N - 1$ zeros of the shifted Legendre polynomials, $P_{N-1}(\nu)$, and the roots being ν_β , $\beta = 1, 2, 3, \dots, N$.

Note that for $\nu = \nu_0$, eqs. (14) become the constraints specified by eqs. (7) [with the convention that $\lambda(\nu_0) = 0$]. The $2N$ linear equations to be solved for a_α and b_α , $\alpha = 0, 1, 2, \dots, N - 1$ are therefore

$$\sum_{\alpha=0}^{N-1} \left[a_\alpha B_{\alpha\beta} + b_\alpha A_{\alpha\beta} e^{-\frac{\Delta}{\nu_\beta}} \right] = R_{1\beta}(a, b, \Delta) \quad (15a)$$

$$\sum_{\alpha=0}^{N-1} \left[a_\alpha A_{\alpha\beta} e^{-\frac{\Delta}{\nu_\beta}} + b_\alpha B_{\alpha\beta} \right] = R_{2\beta}(a, b, \Delta) \quad (15b)$$

where

$$A_{\alpha\beta} = A_\alpha(\nu_\beta) \quad (15c)$$

$$B_{\alpha\beta} = B_\alpha(\nu_\beta) \quad (15d)$$

$$R_{1\beta}(a, b, \Delta) = R_1(\nu_\beta, a, b, \Delta) \quad (15e)$$

$$R_{2\beta}(a, b, \Delta) = R_2(\nu_\beta, a, b, \Delta). \quad (15f)$$

From experience, it has been found that eqs. (13), for the reflected and transmitted fluxes respectively, do not provide exceptionally accurate results, especially near $\mu = 0$. Thus C. E. Siewert in [1] has developed a post-processor which provides a more accurate evaluation. This expression is derived with minor modification by noting that from eq. (14d)

$$B_\alpha(\nu) = \lambda^*(\nu)\psi_\alpha(\nu) - B_\alpha^*(\nu) \quad (16a)$$

where

$$\lambda^*(\nu) = 1 + \frac{c\nu}{2} \ln \frac{\nu}{1+\nu} \quad (16b)$$

$$B_\alpha^*(\nu) = \frac{c}{2} \int_0^1 d\mu \left(\frac{\mu\psi_\alpha(\mu) - \nu\psi_\alpha(\nu)}{\mu - \nu} \right). \quad (16c)$$

When eq. (16a) is introduced into eqs. (14), there results

$$\begin{aligned} \Phi(a, -\mu) = & F_R(\mu)e^{-\frac{\Delta}{\mu}} + \\ & + \frac{c}{2\lambda^*(\mu)} \left\{ R_1(\mu, a, b, \Delta) + \sum_{\alpha=0}^{N-1} [a_\alpha B_\alpha^*(\mu) - b_\alpha A_\alpha^*(\mu)] \right\} \end{aligned} \quad (17a)$$

$$\begin{aligned} \Phi(b, \mu) = & F_L(\mu)e^{-\frac{\Delta}{\mu}} + \\ & + \frac{c}{2\lambda^*(\mu)} \left\{ R_2(\mu, a, b, \Delta) + \sum_{\alpha=0}^{N-1} [b_\alpha B_\alpha^*(\mu) - a_\alpha A_\alpha^*(\mu)] \right\} \end{aligned} \quad (17b)$$

with

$$A_\alpha^*(\mu) \equiv A_\alpha(\mu)e^{-\frac{\Delta}{\mu}}. \quad (17c)$$

Singular integral equations can also be obtained for the flux at a position x in the slab interior by first letting $a = x$ in eq. (12a) to give

$$\begin{aligned} \lambda(\nu)\Phi(x, -\nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} \Phi(x, -\mu) + \\ + \frac{c}{2} e^{-\frac{b-\nu}{\nu}} \int_0^1 d\mu \frac{\mu}{\mu + \nu} \Phi(b, \mu) = \frac{c}{2} \int_0^1 d\mu \frac{\mu}{\mu + \nu} \Phi(x, \mu) + \\ + e^{-\frac{b-\nu}{\nu}} \left\{ \lambda(\nu)F_R(\nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} F_R(\mu) \right\} + \frac{c^{\frac{1}{2}}}{2\nu} S^*(\nu, x, b) \end{aligned} \quad (18a)$$

and letting $b = x$ in eq. (12b) to give

$$\begin{aligned} \lambda(\nu)\Phi(x, \nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} \Phi(x, \mu) + \\ + \frac{c}{2} e^{-\frac{x-a}{\nu}} \int_0^1 d\mu \frac{\mu}{\mu + \nu} \Phi(a, -\mu) = \frac{c}{2} \int_0^1 d\mu \frac{\mu}{\mu + \nu} \Phi(x, -\mu) + \\ + e^{-\frac{x-a}{\nu}} \left\{ \lambda(\nu)F_L(\nu) - \frac{c}{2} \mathcal{P} \int_0^1 d\mu \frac{\mu}{\mu - \nu} F_L(\mu) \right\} + \frac{e^{-\frac{x}{\nu}}}{2\nu} S^*(-\nu, a, x). \end{aligned} \quad (18b)$$

Then, if the approximations

$$\Phi(x, -\mu) = F_R(\mu) e^{-\frac{b-x}{\mu}} + \frac{c}{2} \sum_{\alpha=0}^{N-1} c_\alpha \psi_\alpha(\mu) \quad (19a)$$

$$\Phi(x, \mu) = F_L(\mu) e^{-\frac{x-a}{\mu}} + \frac{c}{2} \sum_{\alpha=0}^{N-1} d_\alpha \psi_\alpha(\mu) \quad (19b)$$

are substituted into eqs. (18) and the result evaluated at $\nu = \nu_\beta$, the following linear set of equations to be solved for c_α and d_α are obtained

$$\sum_{\alpha=0}^{N-1} [c_\alpha B_{\alpha\beta} - d_\alpha A_{\alpha\beta}] = \hat{R}_{1\beta}(a, b, x) - e^{-\frac{b-x}{\nu_\beta}} \sum_{\alpha=0}^{N-1} b_\alpha A_{\alpha\beta} \quad (20a)$$

$$\sum_{\alpha=0}^{N-1} [-c_\alpha A_{\alpha\beta} + d_\alpha B_{\alpha\beta}] = \hat{R}_{2\beta}(a, b, x) - e^{-\frac{x-a}{\nu_\beta}} \sum_{\alpha=0}^{N-1} a_\alpha A_{\alpha\beta} \quad (20b)$$

where

$$\begin{aligned} \hat{R}_1(\nu, a, b, x) \equiv \int_0^1 d\mu \mu \left[F_L(\mu) S(b-x, \mu, \nu) e^{-\frac{x-a}{\mu}} + F_R(\mu) C(b-x, \mu, \nu) \right] + \\ + \frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(\nu, x, b) \end{aligned} \quad (20c)$$

$$\begin{aligned}\tilde{R}_2(\nu, a, b, x) \equiv & \int_0^1 d\mu \mu \left[F_L(\mu) C(x - a, \mu, \nu) + F_R(\mu) S(x - a, \mu, \nu) e^{-\frac{b-x}{\mu}} \right] + \\ & + \frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(-\nu, a, x)\end{aligned}\quad (20d)$$

and

$$R_{1,2\beta}(a, b, x) = R_{1,2}(\nu_\beta, a, b, x). \quad (20e)$$

Since a_α and b_α have previously been determined, this set of equations can be solved for c_α and d_α . The post processing step is

$$\begin{aligned}\Phi(x, -\mu) = & F_R(\mu) e^{-\frac{b-x}{\mu}} + \\ & + \frac{c}{2\lambda^*(\mu)} \left[\tilde{R}_1(\mu, a, b, x) + \sum_{\alpha=0}^{N-1} \left\{ c_\alpha B_\alpha^*(\mu) + \left(d_\alpha - b_\alpha e^{-\frac{b-x}{\mu}} \right) A_\alpha(\mu) \right\} \right]\end{aligned}\quad (21a)$$

$$\begin{aligned}\Phi(x, \mu) = & F_L(\mu) e^{-\frac{x-a}{\mu}} + \\ & + \frac{c}{2\lambda^*(\mu)} \left[\tilde{R}_2(\mu, a, b, x) + \sum_{\alpha=0}^{N-1} \left\{ d_\alpha B_\alpha^*(\mu) + \left(c_\alpha - a_\alpha e^{-\frac{x-a}{\mu}} \right) A_\alpha(\mu) \right\} \right].\end{aligned}\quad (21b)$$

III.B Multiple Slabs

So far, only a single slab has been considered. We now consider a heterogeneous medium composed of a set of slabs bordering a vacuum. On the left vacuum/slab interface, a beam source of strength S_0 and direction μ_0 is assumed to impinge. On the right vacuum/slab interface, there is no source. The connection between each slab is made through the boundary term on the right hand side's of the eqs. (15) and eqs. (20). Three distinct cases can be identified including:

LVS-RSS: left vacuum with a beam source/slab interface, right slab/slab interface,

LSS-RSS: left and right both slab/slab interface, and

LSS-RSV: left slab/slab interface, right slab/vacuum interface.

The particular right hand side's for each interface will now be specified. Since $R_1(\nu, a, b, \Delta) = \tilde{R}_1(\nu, a, b, a)$ and $R_2(\nu, a, b, \Delta) = \tilde{R}_2(\nu, a, b, b)$, only $\tilde{R}_1$ and $\tilde{R}_2$ need be considered in the following description.

1. LVS-RSS

For this case, the slab is the initial slab going from left to right and

$$F_L(\mu) = S_0 \delta(\mu - \mu_0), \quad \mu, \mu_0 > 0$$

$$F_R = \Phi_2(\Delta_1, -\mu), \quad \mu > 0$$

where Δ_1 is the thickness of the first slab and $\Phi_2(\Delta_1, -\mu)$ is the reflected flux from the second slab. Then from eqs. (20c,d) (for $a = 0$ and $b = \Delta_1$)

$$\begin{aligned} \tilde{R}_1(\nu, 0, \Delta_1, x) = & \mu_0 S_0 S(\Delta_1 - x, \mu_0, \nu) e^{-\frac{x}{\mu_0}} + \frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(\nu, x, \Delta_1) + \\ & + \int_0^1 d\mu \mu C(\Delta_1 - x, \mu, \nu) \Phi_2(\Delta_1, -\mu) - e^{-\frac{\Delta_1}{\nu\beta}} \sum_{\alpha=0}^{N-1} b_\alpha A_{\alpha\beta} \end{aligned} \quad (22a)$$

$$\begin{aligned} \tilde{R}_2(\nu, 0, \Delta_1, x) = & \mu_0 S_0 C(x, \mu_0, \nu) + \frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(-\nu, 0, x) + \\ & + \int_0^1 d\mu \mu S(x, \mu, \nu) \Phi_2(\Delta_1, -\mu) e^{-\frac{\Delta_1 - x}{\mu}} - e^{-\frac{x}{\nu\beta}} \sum_{\alpha=0}^{N-1} b_\alpha A_{\alpha\beta}. \end{aligned} \quad (22b)$$

2. LSS-RSS

For slab k , the entering fluxes are (for $a = x_{k-1}$ and $b = x_k$)

$$F_L(\mu) = \Phi_{k-1}(x_{k-1}, \mu) + S_0 e^{-\frac{x_{k-1}}{\mu_0}} \delta(\mu - \mu_0)$$

$$F_R(\mu) = \Phi_{k+1}(x_k, -\mu)$$

where the uncollided component has been explicitly accounted for. Thus

$$\begin{aligned} \tilde{R}_1(\nu, x_{k-1}, x_k, x) = & \mu_0 S_0 S(x_k - x, \mu_0, \nu) e^{-\frac{x}{\mu_0}} + \frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(\nu, x, x_k) + \\ & + \int_0^1 d\mu \mu \left[S(x_{k-1} - x, \mu, \nu) \Phi_{k-1}(x_{k-1}, \mu) e^{-\frac{x - x_{k-1}}{\mu}} + \right. \\ & \left. + C(x_k - x, \mu, \nu) \Phi_{k+1}(x_k, -\mu) \right] - e^{-\frac{x_k - x}{\nu\beta}} \sum_{\alpha=0}^{N-1} b_\alpha A_{\alpha\beta} \end{aligned} \quad (22c)$$

$$\begin{aligned} \tilde{R}_2(\nu, x_{k-1}, x_k, x) = & \mu_0 S_0 C(x - x_{k-1}, \mu_0, \nu) e^{-\frac{x_{k-1}}{\mu_0}} + \frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(-\nu, x_{k-1}, x) + \\ & + \int_0^1 d\mu \mu \left[S(x - x_{k-1}, \mu, \nu) \Phi_{k+1}(x_k, -\mu) e^{-\frac{x_k - x}{\mu}} + \right. \\ & \left. + C(x - x_{k-1}, \mu, \nu) \Phi_{k-1}(x_{k-1}, \mu) \right] - e^{-\frac{x - x_{k-1}}{\nu\beta}} \sum_{\alpha=0}^{N-1} a_\alpha A_{\alpha\beta}. \end{aligned} \quad (22d)$$

3. Lss-Rsv

This is the case for the right most slab, designated slab K, where

$$F_L(\mu) = \Phi_{K-1}(x_{K-1}, \mu) + S_0 e^{-\frac{x_{K-1}}{\mu_0}} \delta(\mu - \mu_0)$$

$$F_R(\mu) = 0$$

Therefore (for $a = x_{K-1}$ and $b = x_K$)

$$\begin{aligned} \tilde{R}_1(\nu, x_{K-1}, x_K, x) = & \mu_0 S_0 S(x_K - x, \mu_0, \nu) e^{-\frac{x}{\mu_0}} + \frac{e^{\frac{x}{\nu}}}{c\nu} S^*(\nu, x, x_K) + \\ & + \int_0^1 d\mu \mu S(x_K - x, \mu, \nu) \Phi_{K-1}(x_{K-1}, \mu) e^{-\frac{x-x_{K-1}}{\mu}} - \\ & - e^{-\frac{x_{K-1}}{\nu}} \sum_{\alpha=0}^{N-1} b_\alpha A_{\alpha\beta} \end{aligned} \quad (22e)$$

$$\begin{aligned} \tilde{R}_2(\nu, x_{K-1}, x_K, x) = & \mu_0 S_0 C(x - x_{K-1}, \mu_0, \nu) e^{-\frac{x_{K-1}}{\mu_0}} + \\ & + \frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(-\nu, x_{K-1}, x) + \int_0^1 d\mu \mu C(x - x_{K-1}, \mu, \nu) \Phi_{K-1}(x_{K-1}, \mu) - \\ & - e^{-\frac{x-x_{K-1}}{\nu}} \sum_{\alpha=0}^{N-1} a_\alpha A_{\alpha\beta}. \end{aligned} \quad (22f)$$

III.C Uniform Source

A useful source distribution to assume is one that varies exponentially across each slab, $S(x) = S_k e^{-\alpha_k x}$. Thus, the required S^* for slab k is simply given by

$$\begin{aligned} S^*(\nu, a, b) &= S_k \int_a^b d\tau e^{-(\alpha_k + \frac{1}{\nu})\tau} \\ &= \frac{S_k \nu}{1 + \alpha_k \nu} \left[e^{-(\alpha_k + \frac{1}{\nu})a} - e^{-(\alpha_k + \frac{1}{\nu})b} \right]. \end{aligned}$$

The appropriate source contributions for R_1 , R_2 , $\tilde{R}_1$, and $\tilde{R}_2$ are for R_1 :

$$\frac{e^{\frac{x}{\nu}}}{c\nu} S^*(\nu, a, b) = \frac{S_k e^{-\alpha_k a}}{c(1 + \alpha_k \nu)} \left[1 - e^{-(\alpha_k + \frac{1}{\nu})\Delta_k} \right] \quad (23a)$$

for R_2 :

$$\frac{e^{-\frac{b}{\nu}}}{c\nu} S^*(-\nu, a, b) = \frac{S_k e^{-\alpha_k a}}{c(1 - \alpha_k \nu)} \left[e^{-\frac{\Delta_k}{\nu}} - e^{-\alpha_k \Delta_k} \right] \quad (23b)$$

for $\tilde{R}_1$:

$$\frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(\nu, x, x_k) = \frac{S_k e^{-\alpha_k x}}{c(1 + \alpha_k \nu)} \left[1 - e^{-(x_k - x)(\alpha_k + \frac{1}{\nu})} \right] \quad (23c)$$

for $\tilde{R}_2$:

$$\frac{e^{-\frac{x}{\nu}}}{c\nu} S^*(\nu, x_{k-1}, x) = \frac{S_k e^{-\alpha_k x_{k-1}}}{c(1 - \alpha_k \nu)} \left[e^{-\frac{x - x_{k-1}}{\nu}} - e^{-\alpha_k (x - x_{k-1})} \right]. \quad (23d)$$

IV Numerical Implementation and Demonstration

IV.A Numerical Considerations

Several decisions concerning the numerical method associated with the F_N method must be made, including:

- numerical quadrature
- zero search
- choice of ν_β
- choice of basis functions
- matrix inversion
- iteration strategy.

All integrals required for the matrix elements and inhomogeneous terms will be performed using a shifted Legendre-Gauss quadrature scheme. While the numerical integration is straight forward for A_n , $\tilde{R}_1$, and $\tilde{R}_2$, it is not for B_n . This is a consequence of the principle

value integration. However, upon regularization of the integral as performed for the post processing step, we need only to evaluate $B_{\alpha}^*(\nu)$ given by eq. (16c) which no longer presents a difficulty. In addition, to allow for some flexibility and numerical adjustment, a different quadrature order than for the matrix elements is allowed for the integrals over the boundary fluxes contained in the inhomogeneous terms connecting the slabs (R_1 and R_2). It should be noted that the evaluation of the matrix elements differ from the recurrence relations used originally in [1]. Comparable accuracy using quadratures however can be achieved.

A zero search is performed to determine ν_0 from $\Lambda(\nu_0) = 0$. A simple bisection procedure with zero bracketing and refinement is employed.

The most convenient choice for ν_{β} , $\beta = 1, 2, \dots, N$ are the $N - 1$ zeros of the shifted Legendre polynomials $P_{N-1}^*(\nu)$ on $[0, 1]$. Other choices are also available such as uniform spacing and the zeros of shifted Chebyshev polynomials $T_{N-1}^*(\nu)$; however no significant difference has been observed with these choices. The choice of the basis functions can be crucial to the proper operation of the algorithm. In the past, both the monomials μ^{α} and shifted Legendre polynomials $P_{\alpha}^*(\mu)$ have been used. Because of their ill-conditioned behavior, however, monomials have been abandoned. In addition to the above choices, several others are possible, including shifted monomials $(2\mu^{\gamma} - 1)^{\alpha}$ and modified Legendre polynomials $P_{\alpha}(2\mu^{\gamma} - 1)$ where γ is a free parameter. From experience, best results have been found for $\gamma \approx 0.75$.

The linear system of equations generated by the F_N algorithm for both the boundary and interior fluxes are generally of the form ($\beta = 0, 1, \dots, N - 1$)

$$\sum_{\alpha=0}^{N-1} [c_{\alpha} K_{\alpha\beta} + f_{\alpha} H_{\alpha\beta}] = \chi_{1\beta} \quad (24a)$$

$$\sum_{\alpha=0}^{N-1} [c_{\alpha} H_{\alpha\beta} + f_{\alpha} K_{\alpha\beta}] = \chi_{2\beta} \quad (24b)$$

Adding and subtracting these equations yields

$$\sum_{\alpha=0}^{N-1} [(c_{\alpha} + f_{\alpha})(K_{\alpha\beta} + H_{\alpha\beta})] = \chi_{1\beta} + \chi_{2\beta} \quad (25a)$$

$$\sum_{\alpha=0}^{N-1} [(c_{\alpha} - f_{\alpha})(K_{\alpha\beta} - H_{\alpha\beta})] = \chi_{1\beta} - \chi_{2\beta} \quad (25b)$$

These equations are solved for $g_{\alpha}^{+} = e_{\alpha} + f_{\alpha}$ and $g_{\alpha}^{-} = e_{\alpha} - f_{\alpha}$ using the LU decomposition [2], then

$$e_{\alpha} = \frac{1}{2} (g_{\alpha}^{+} + g_{\alpha}^{-}) \quad (26a)$$

$$f_{\alpha} = \frac{1}{2} (g_{\alpha}^{+} - g_{\alpha}^{-}) \quad (26b)$$

The following iteration strategy has been adopted in order to ensure convergence of the angular flux. The order N of the approximation is increased until the fluxes on the slab boundaries have converged to a desired relative error. For more than one slab, an additional inner iteration at each N is imposed since each slab is considered separately and the entering fluxes are not known *a priori*. The inner iteration, essentially additional cycles through the slabs, are performed with the boundary fluxes updated after each cycle. Upon convergence of the boundary fluxes, the (N) iteration for the interior fluxes is begun at the order of convergence of the boundary fluxes in order to avoid unproductive lower (N) order iterations.

In addition to the angular flux, the reflection and transmission coefficients,

$$A^{*} = \int_0^1 d\mu \mu \phi(a, -\mu) / S_0 \quad (27a)$$

$$B^{*} = \int_0^1 d\mu \mu \phi(b, \mu) / S_0, \quad (27b)$$

are desired. Since the angular flux at the boundaries (a and b) are known at Gauss-Legendre quadrature points, the integrations in eqs. (27) can be easily performed.

IV.B Demonstration

In order to test the accuracy of the MSLAB code and test the algorithms for proper coding, several trials have been performed, the results of which will now be reported.

Trial 1 involves the verification of the accuracy of a single slab of width 1 and $c = 0.9$ by setting the quadrature order to 30 and varying the desired relative error ϵ . Table 1 shows

the reflected and transmitted fluxes for $\epsilon = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$ with the results for the last case identical to those of [1]. As can be observed, the results are always within the prescribed accuracy giving confidence in the coding of the algorithm. As ϵ is further reduced to 10^{-8} , the results remain invariant. An F_N approximation of $N = 22$ was required to achieve an accuracy of 10^{-5} . Table 2 is similar to Table 1 for $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$ for the interior point $x = 0.5$. At $\epsilon = 10^{-6}$ the results are identical to those of [1].

In the second trial the quadrature order, L_m , was varied for the same problem with $\epsilon = 10^{-5}$. For $L_m = 16$, the fluxes have converged to the results of [1] indicating that a rather low order quadrature will suffice. Table 3 contains these results. The third trial was designed to test the multislabs aspect of the F_N algorithm. For this case, the slab of $\Delta = 1$ and $c = 0.9$ was partitioned into 6 slabs each with $c = 0.9$ and of widths:

Slab Number	Δ
1	0.05
2	0.05
3	0.1
4	0.3
5	0.25
6	0.25

The choice of these widths was dictated by the results of [1] which contained evaluations at $x = 0.05, 0.1, 0.5, 0.75$, and 1.0 . Three inner iterations for each N were imposed with $L_m = 20$ and $\epsilon = 10^{-5}$. The quadrature order, L_{m3} , for the integrals involving boundary fluxes was varied (starting at 5 and ending at 25 with intervals of 5) with the reflected flux, at $x = 0.2$, and the transmitted flux, at $x = 0.5$, displayed in Table 4. In comparison with the results of [1] for $x = 0.5$, the algorithm results are off at most one unit in the fifth place for $L_{m3} \geq 15$.

The final trial investigated how the computational time and accuracy was influenced by the number of inner iterations (LIT). Table 5 shows that with zero inner iteration ($LIT=1$), the 6 slab case did not converge for the $N=60$ approximation. With 1 or more iterations ($LIT \geq 2$), the fluxes converge for $N=54$ with no improvement as LIT increased above 2. The computational time on a CONVEX mainframe relative to zero iterations (CPU_0) is also given, indicating no advantage to require more than 2 inner iterations.

Table 1: Reflected and Transmitted Fluxes for Variation of ϵ

$$\Delta = 1.0, c = 0.9, L_m = 30, \text{IBR} = 4$$

	$\epsilon = 10^{-2}$ and $N = 14$		$\epsilon = 10^{-3}$ and $N = 14$	
μ	Reflected Flux	Transmitted Flux	Reflected Flux	Transmitted Flux
1.0	2.10001E-01	1.90264E-01	2.10001E-01	1.90264E-01
0.9	2.23885E-01	2.00704E-01	2.23885E-01	2.00704E-01
0.8	2.39472E-01	2.11883E-01	2.39472E-01	2.11883E-01
0.7	2.56973E-01	2.23633E-01	2.56973E-01	2.23633E-01
0.6	2.76545E-01	2.35537E-01	2.76545E-01	2.35537E-01
0.5	2.98151E-01	2.46678E-01	2.98151E-01	2.46678E-01
0.4	3.21267E-01	2.55202E-01	3.21267E-01	2.55202E-01
0.3	3.44242E-01	2.57585E-01	3.44242E-01	2.57585E-01
0.2	3.63302E-01	2.48316E-01	3.63302E-01	2.48316E-01
0.1	3.72679E-01	2.24054E-01	3.72679E-01	2.24054E-01
1e-10	3.59370E-01	1.86136E-01	3.59370E-01	1.86136E-01
	$\epsilon = 10^{-4}$ and $N = 16$		$\epsilon = 10^{-5}$ and $N = 22$	
μ	Reflected Flux	Transmitted Flux	Reflected Flux	Transmitted Flux
1.0	2.10001E-01	1.90264E-01	2.10001E-01	1.90265E-01
0.9	2.23885E-01	2.00704E-01	2.23885E-01	2.00704E-01
0.8	2.39472E-01	2.11883E-01	2.39472E-01	2.11883E-01
0.7	2.56974E-01	2.23634E-01	2.56974E-01	2.23633E-01
0.6	2.76545E-01	2.35536E-01	2.76545E-01	2.35536E-01
0.5	2.98151E-01	2.46679E-01	2.98151E-01	2.46679E-01
0.4	3.21267E-01	2.55202E-01	3.21267E-01	2.55202E-01
0.3	3.44241E-01	2.57582E-01	3.44241E-01	2.57582E-01
0.2	3.63306E-01	2.48327E-01	3.63306E-01	2.48328E-01
0.1	3.72670E-01	2.24032E-01	3.72669E-01	2.24033E-01
1E-10	3.59371E-01	1.86138E-01	3.59371E-01	1.86138E-01

Table 2: Reflected and Transmitted Fluxes for ϵ Variation for Interior Point

$$x = 0.5, \Delta = 1.0, c = 0.9, L_m = 30, \text{IBR} = 4$$

μ	$\epsilon = 10^{-2}$		$\epsilon = 10^{-4}$	
	Reflected Flux	Transmitted Flux	Reflected Flux	Transmitted Flux
1.0	1.08052E-01	1.43159E-01	1.08052E-01	1.43159E-01
0.9	1.17325E-01	1.55000E-01	1.17325E-01	1.55000E-01
0.8	1.28289E-01	1.68882E-01	1.28289E-01	1.68882E-01
0.7	1.41428E-01	1.85331E-01	1.41428E-01	1.85331E-01
0.6	1.57408E-01	2.05035E-01	1.57408E-01	2.05035E-01
0.5	1.77150E-01	2.28845E-01	1.77150E-01	2.28845E-01
0.4	2.01882E-01	2.57648E-01	2.01882E-01	2.57648E-01
0.3	2.32986E-01	2.91656E-01	2.32986E-01	2.91656E-01
0.2	2.70683E-01	3.27322E-01	2.70683E-01	3.27322E-01
0.1	3.08155E-01	3.47036E-01	3.08155E-01	3.47036E-01
1E-10	3.32748E-01	3.32748E-01	3.32748E-01	3.32748E-01

μ	$\epsilon = 10^{-6}$	
	Reflected Flux	Transmitted Flux
1.0	1.08052E-01	1.43159E-01
0.9	1.17325E-01	1.55000E-01
0.8	1.28289E-01	1.68882E-01
0.7	1.41428E-01	1.85331E-01
0.6	1.57408E-01	2.05035E-01
0.5	1.77150E-01	2.28845E-01
0.4	2.01882E-01	2.57648E-01
0.3	2.32986E-01	2.91657E-01
0.2	2.70683E-01	3.27320E-01
0.1	3.08155E-01	3.47040E-01
1E-10	3.32750E-01	3.32750E-01

Table 3: Reflected and Transmitted Fluxes from Lm Variation

$$\Delta = 1.0, c = 0.9, \epsilon = 10^{-5}, \text{IBR} = 4$$

	Lm = 2		Lm = 4	
μ	Reflected Flux	Transmitted Flux	Reflected Flux	Transmitted Flux
1.0	2.11696E-01	1.92163E-01	2.09973E-01	1.90221E-01
0.9	2.25496E-01	2.02572E-01	2.23856E-01	2.00656E-01
0.8	2.41032E-01	2.13761E-01	2.39441E-01	2.11830E-01
0.7	2.58512E-01	2.25564E-01	2.56940E-01	2.23573E-01
0.6	2.78096E-01	2.37571E-01	2.76509E-01	2.35468E-01
0.5	2.99759E-01	2.48878E-01	2.98111E-01	2.46601E-01
0.4	3.22988E-01	2.57634E-01	3.21222E-01	2.55112E-01
0.3	3.46144E-01	2.60292E-01	3.44190E-01	2.57478E-01
0.2	3.65427E-01	2.51228E-01	3.63252E-01	2.48219E-01
0.1	3.74872E-01	2.26683E-01	3.72627E-01	2.23955E-01
1E-10	3.61469E-01	1.88096E-01	3.59387E-01	1.86136E-01
	Lm = 8		Lm = 16	
μ	Reflected Flux	Transmitted Flux	Reflected Flux	Transmitted Flux
1.0	2.10001E-01	1.90264E-01	2.10001E-01	1.90265E-01
0.9	2.23885E-01	2.00704E-01	2.23885E-01	2.00704E-01
0.8	2.39472E-01	2.11883E-01	2.39472E-01	2.11883E-01
0.7	2.56973E-01	2.23633E-01	2.56974E-01	2.23633E-01
0.6	2.76545E-01	2.35536E-01	2.76545E-01	2.35536E-01
0.5	2.98151E-01	2.46678E-01	2.98151E-01	2.46679E-01
0.4	3.21267E-01	2.55201E-01	3.21267E-01	2.55202E-01
0.3	3.44240E-01	2.57581E-01	3.44241E-01	2.57582E-01
0.2	3.63306E-01	2.48326E-01	3.63306E-01	2.48328E-01
0.1	3.72669E-01	2.24032E-01	3.72669E-01	2.24033E-01
1E-10	3.59374E-01	1.86139E-01	3.59371E-01	1.86138E-01

Table 4: Reflected and Transmitted Fluxes from Lm3 Variation

Six Slabs, $c = 0.9$, $Lm = 20$, $\epsilon = 10^{-5}$, $IBR = 4$

μ	Lm3 = 5		Lm3 = 10	
	Reflected Flux ($x = 0.2$)	Transmitted Flux ($x = 0.5$)	Reflected Flux ($x = 0.2$)	Transmitted Flux ($x = 0.5$)
1.0	1.73095E-01	1.43249E-01	1.73230E-01	1.43151E-01
0.9	1.85873E-01	1.55093E-01	1.86017E-01	1.54991E-01
0.8	2.00532E-01	1.68977E-01	2.00687E-01	1.68872E-01
0.7	2.17448E-01	1.85428E-01	2.17615E-01	1.85322E-01
0.6	2.37046E-01	2.05131E-01	2.37227E-01	2.05025E-01
0.5	2.59741E-01	2.28934E-01	2.59935E-01	2.28833E-01
0.4	2.85724E-01	2.57721E-01	2.85931E-01	2.57635E-01
0.3	3.14366E-01	2.91694E-01	3.14577E-01	2.91644E-01
0.2	3.42795E-01	3.27284E-01	3.42978E-01	3.27307E-01
0.1	3.64943E-01	3.46957E-01	3.64926E-01	3.47028E-01
1E-10	3.77788E-01	3.32759E-01	3.77350E-01	3.32744E-01
μ	Lm3 = 15		Lm3 = 20	
	Reflected Flux ($x = 0.2$)	Transmitted Flux ($x = 0.5$)	Reflected Flux ($x = 0.2$)	Transmitted Flux ($x = 0.5$)
1.0	1.73234E-01	1.43159E-01	1.73234E-01	1.43159E-01
0.9	1.86021E-01	1.55000E-01	1.86021E-01	1.55000E-01
0.8	2.00691E-01	1.68882E-01	2.00691E-01	1.68882E-01
0.7	2.17620E-01	1.85331E-01	2.17620E-01	1.85331E-01
0.6	2.37232E-01	2.05035E-01	2.37232E-01	2.05035E-01
0.5	2.59941E-01	2.28845E-01	2.59941E-01	2.28845E-01
0.4	2.85937E-01	2.57648E-01	2.85937E-01	2.57647E-01
0.3	3.14583E-01	2.91657E-01	3.14583E-01	2.91657E-01
0.2	3.42984E-01	3.27320E-01	3.42984E-01	3.27320E-01
0.1	3.64932E-01	3.47040E-01	3.64932E-01	3.47040E-01
1E-10	3.77346E-01	3.32751E-01	3.77347E-01	3.32751E-01

Table 4: Reflected and Transmitted Fluxes from Lm3 Variation (continued)

Six Slabs, $c = 0.9$, $Lm = 20$, $\epsilon = 10^{-5}$, $IBR = 4$

	Lm3 = 25	
μ	Reflected Flux ($x = 0.2$)	Transmitted Flux ($x = 0.5$)
1.0	1.73233E-01	1.43159E-01
0.9	1.86021E-01	1.55000E-01
0.8	2.00691E-01	1.68881E-01
0.7	2.17620E-01	1.85331E-01
0.6	2.37232E-01	2.05035E-01
0.5	2.59941E-01	2.28844E-01
0.4	2.85937E-01	2.57647E-01
0.3	3.14583E-01	2.91657E-01
0.2	3.42984E-01	3.27320E-01
0.1	3.64932E-01	3.47040E-01
1E-10	3.77346E-01	3.32750E-01

Table 5: Variation of CPU Time with LIT

LIT	CPU / CPU ₀	Comments
1	1.00	Did not converge for N = 60
2	1.83	Converged for N = 54
3	2.01	Converged for N = 54
4	2.24	Converged for N = 54

V Code Operation

This section provides the operational details for the MSLAB code, including

- a detailed subprogram level flow chart along with an explanation of the purpose of each subprogram,
- explanation of the input deck and output using three examples to regenerate data of [1], and
- detailed programming notes to ensure proper use.

V.A Flow Chart

A list of the subprogram names and the functions performed by each follows:

MULTISLAB: Main driver program for the F_N Method in a heterogeneous medium.

S1: Sets boundary conditions and interface type for each slab.

INT0: Determines integrals over boundary fluxes required for the reflection and transmission coefficients.

POST: Calls post-processing routines for boundary and interior fluxes.

PP2: Post-processing routine for boundary fluxes.

PP2Z: Post-processing routine for interior fluxes.

FNX: Determines interior fluxes.

SOLUTB: Calls for the matrix inversion of the F_N collocation equations and calculates the reflection and transmission coefficients.

MELB: Determines matrix elements and inhomogeneous terms.

INTR: Evaluation of integrals over boundary fluxes for inhomogeneous terms required in collocation equations for boundary fluxes.

INTRZ: Evaluation of integrals over boundary fluxes for inhomogeneous terms required in collocation equations for interior fluxes.

BASE: Evaluation of basis functions.

DBASE: Used when $\mu = \nu$ ($AM=V$) in determination of matrix element B_α .

BiSEC1: Performs zero search via Bi-Section method.

GAULEG: Determination of Gauss-Legendre abscissae and weights from [2].

ALAME: Dispersion relation.

TEXP: Normal FORTRAN exponential function with underflow set to zero.

MATINV: Matrix inversion routine that uses LU decomposition.

LUDCMP: Performs LU decomposition from [2].

LUBKSB: Uses back substitution to find answer from [2].

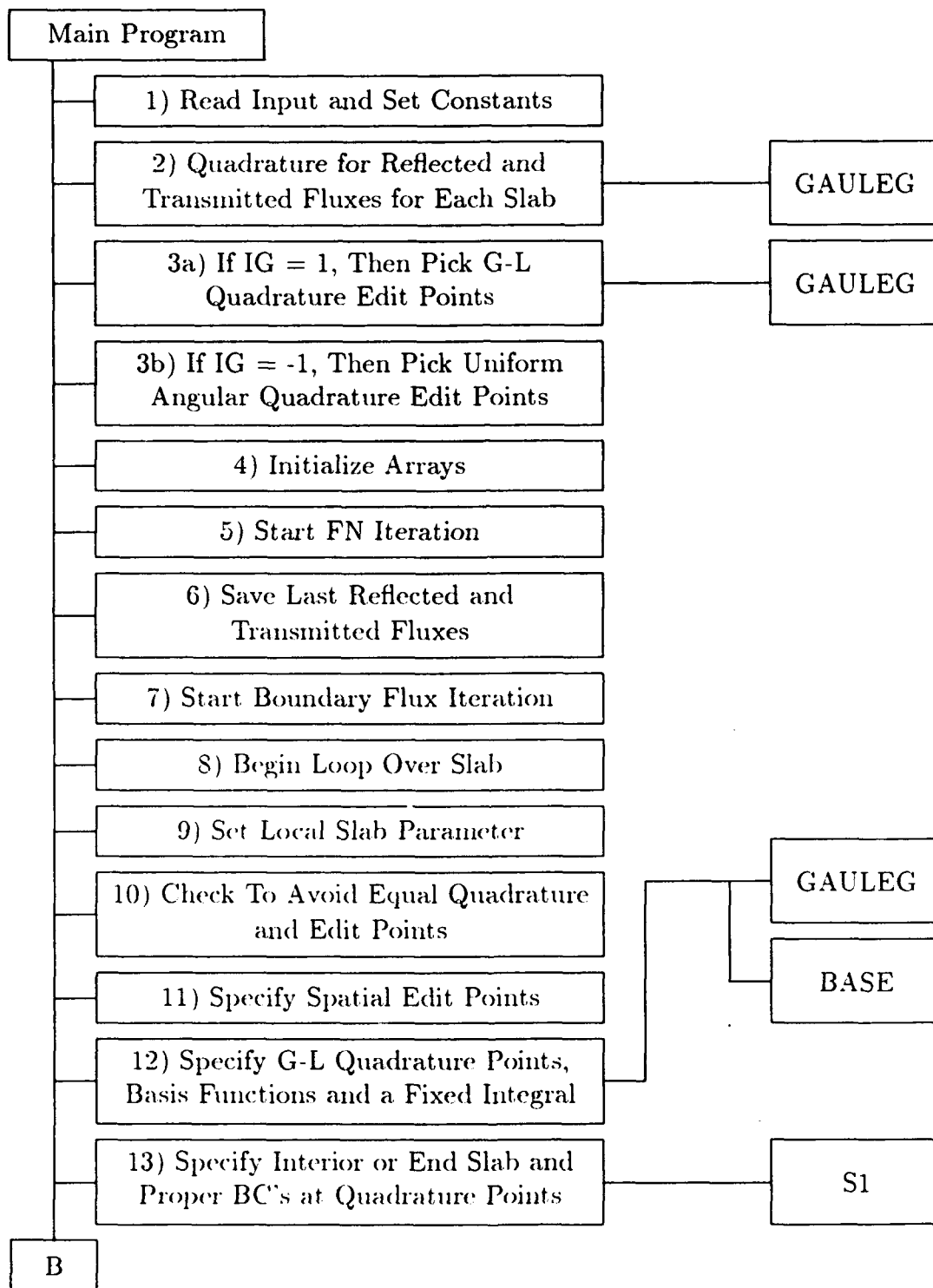
SFCN: Function $S(x,y)$ [eq. (14h)].

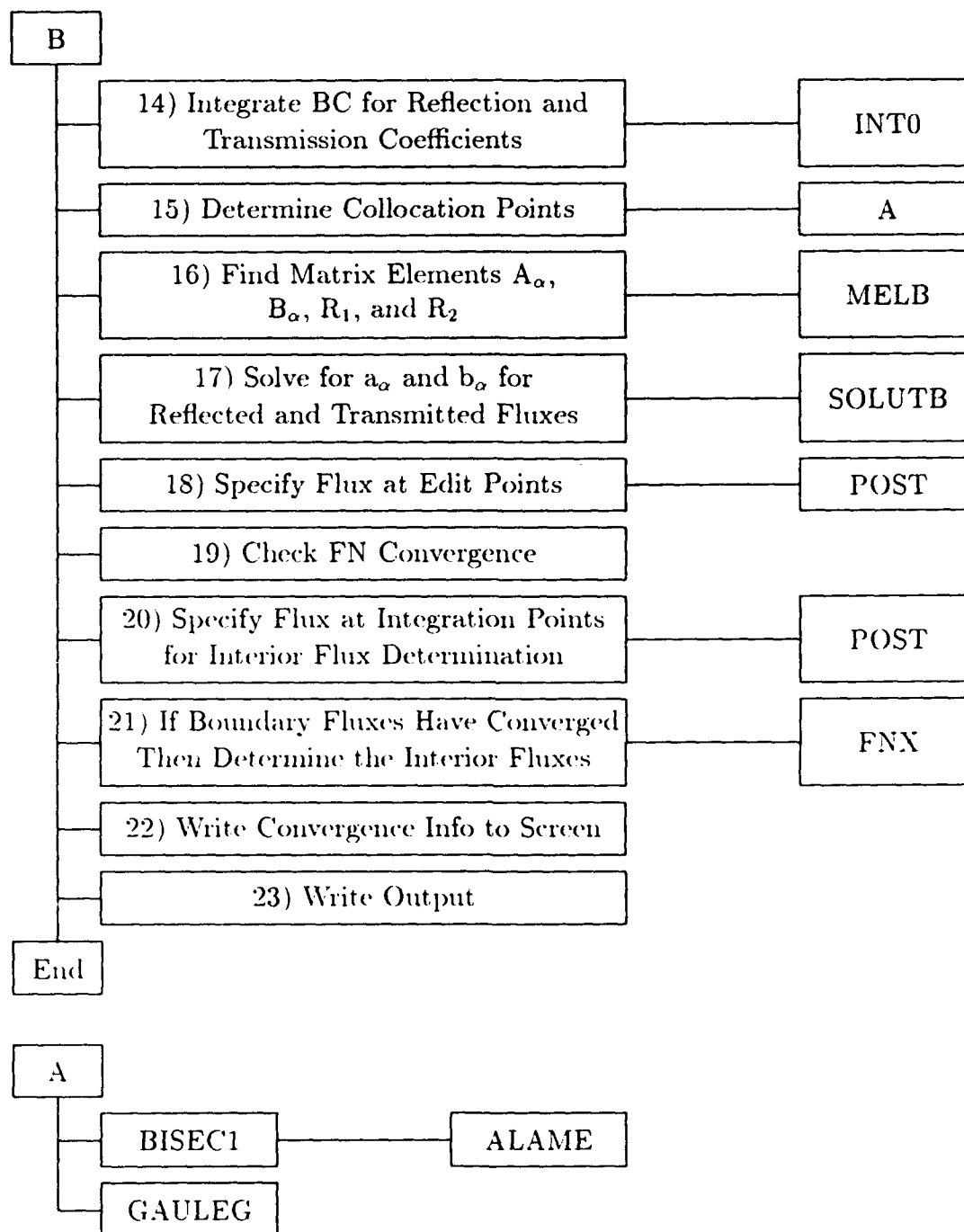
CFCN: Function $C(x,y)$ [eq. (14g)].

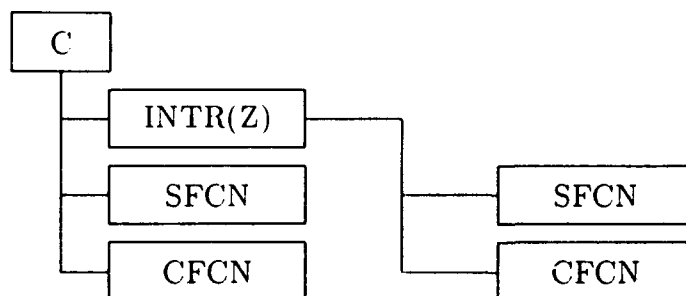
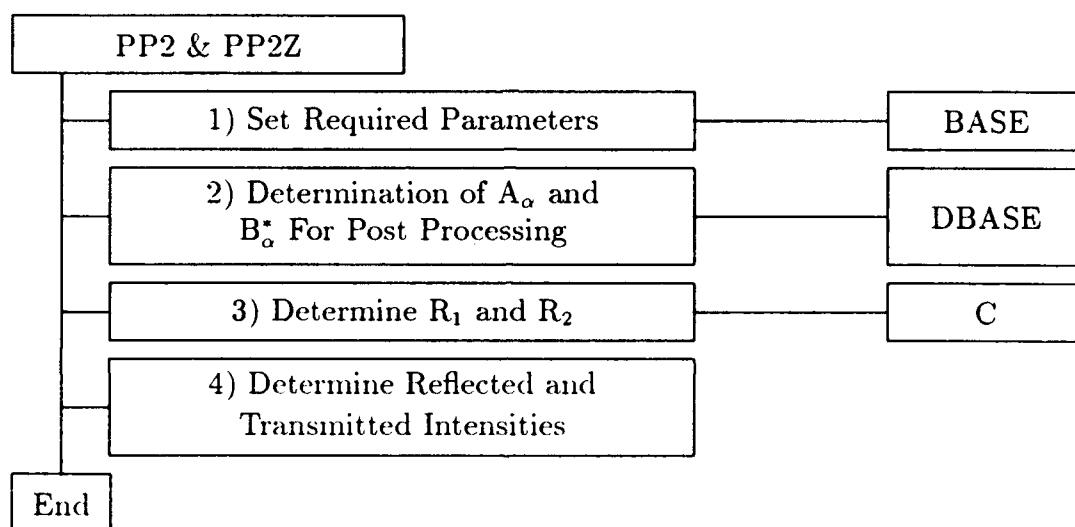
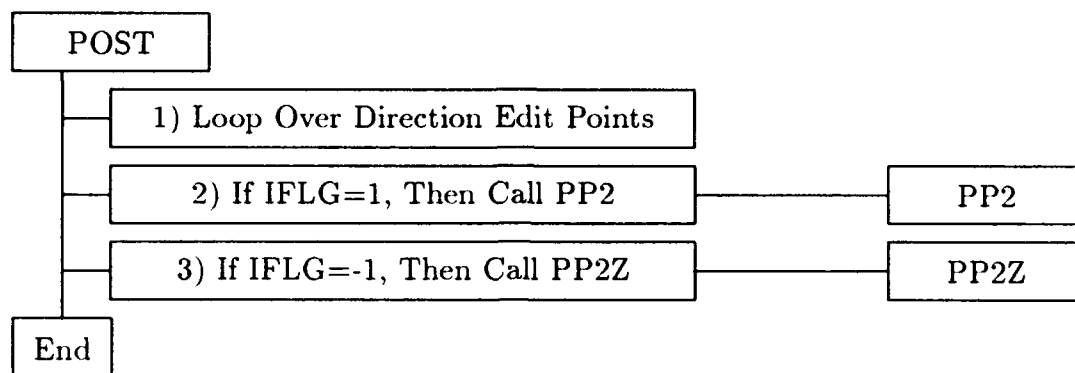
MCOPY: Copies from one 2-D matrix to another.

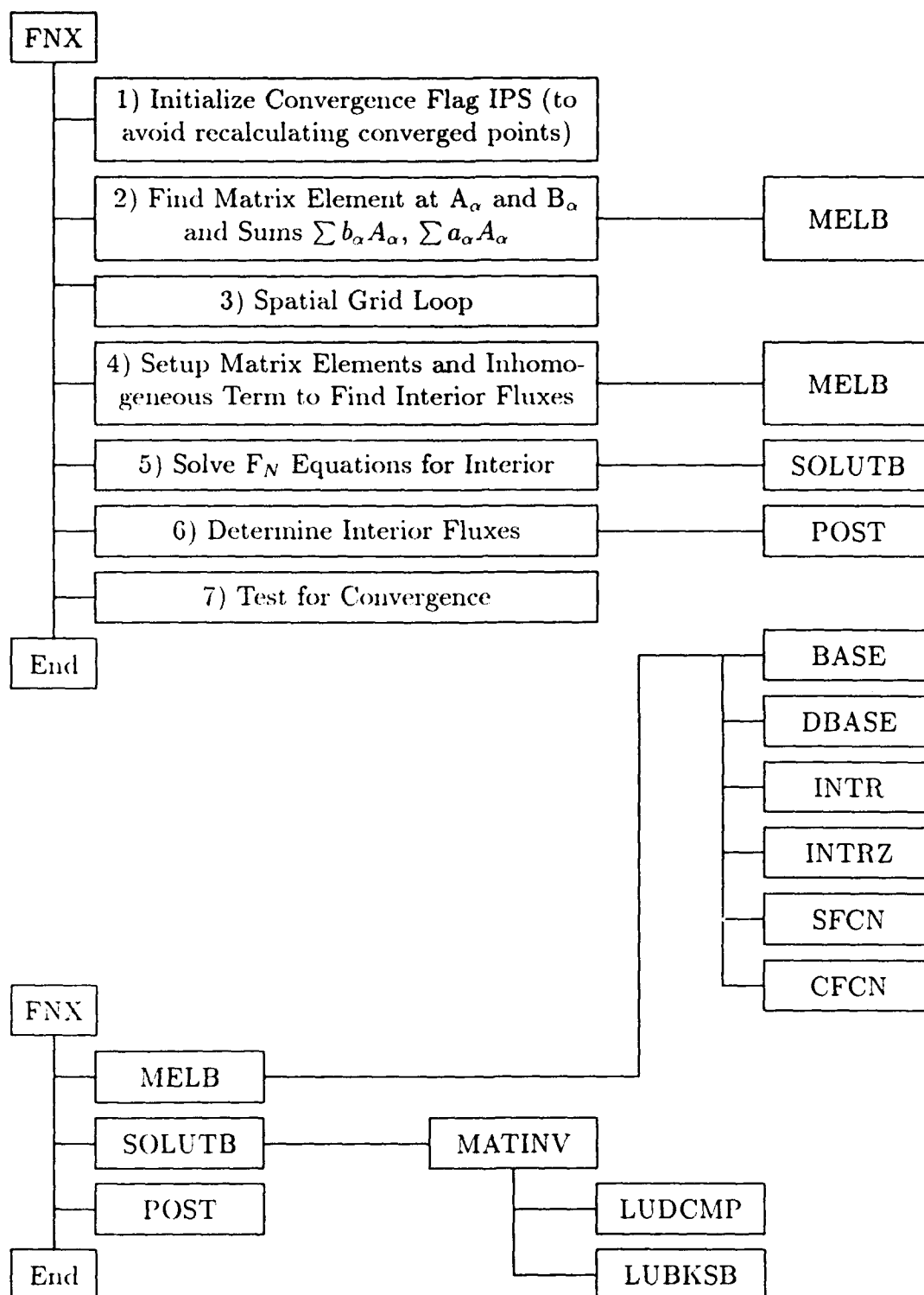
ICOPY: Copies from integer vector to another.

All subprograms listed above are Subroutines except SFCN, CFCN, and TEXP which are Functions.









V.B Input Description

The input deck required to execute this program is explained in this section. Suggested values and limitations are noted in parenthesis. The following text is also contained in the source code.

LINE 1 NS NUMBER OF SLABS (<INS)
 LM3 GAUSS-LEGENDRE INTEGRATION ORDER FOR EVALUATION
 OF INTEGRALS OVER BOUNDARY FLUXES (<ILM)
 LIT NUMBER OF PASSES THROUGH SLABS FOR EACH FN APPROX. (2)

LINE 2 L1 INITIAL N OF FN APPROXIMATION
 L2 FINAL N OF FN APPROXIMATION (<NP)
 L12D INCREMENT BETWEEN APPROXIMATIONS (2)

LINE 3 MM1 NUMBER OF DIRECTION EDIT POINTS (<IMM1)
 IG -1 USE G-L ANGULAR EDIT POINTS (ORDER MM1)
 1 SPECIFY UNIFORM ANGULAR EDIT POINTS

NOTE: IF IG IS 1 ENTER

LINE 4 AN0 FIRST EDIT DIRECTION
 AN1 LAST EDIT DIRECTION

LINE 5 ERR1 DESIRED RELATIVE ERROR (1.0E-04)

LINE 6 AM0 SOURCE DIRECTION
 S0 INTENSITY OF BOUNDARY SOURCE

NOTE: ENTER LINE 7 FOR EACH OF THE NS SLABS

LINE 7 WSR SINGLE SCATTER ALBEDO
 DR SLAB WIDTH
 IBR 1 SHIFTED LEGENDRE POLYNOMIAL BASIS FCNS (PL(2*X-1))
 2 MONOMIAL BASIS FCNS (X**(XAR*L))
 3 SHIFTED MONOMIAL BASIS FCNS ((2*(X**XAR)-1)**L)
 4 SHIFTED LEGENDRE POLYNOMIAL BASIS FCNS
 (PL(U), U=2*(X**XAR)-1)
 LMR G-L QUADRATURE ORDER FOR MATRIX ELEMENTS (<ILM)
 XAR PARAMETER FOR BASIS FCNS OR GAMMA WHEN IBR
 IS 3 OR 4 (0.75)
 S1R EXPONENTIAL VOLUME SOURCE STRENGTH
 ALP1R EXPONENT FOR VOLUME SOURCE

AD-A257 986

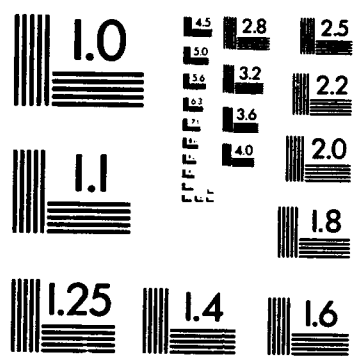
METHODS DEVELOPMENT FOR ELECTRON TRANSPORT (U) CALSPAN
UB RESEARCH CENTER BUFFALO NY B D GANAPOL APR 92
RL**-TR-92-48

2/2

UNCLASSIFIED

NL

END
FILMED
LPA
DTIC



V.B.1 Sample Problem Input

The sample problems are designed to reproduce the data contained in Table 2 of [1]. With the MSLAB program, there are several ways to generate the data. Three different input decks are shown and explained below.

Input deck 1:

- One Slab of width (Δ) 1.0.
- Evenly spaced interior spatial edit points at a spacing of 0.05.

```
1 16 3
11 41 2
10 1
0.0 1.0
1.0E-05
1.0 0.5
0.9 1.0 4 30 0.75 0.0 0.0 20
```

Input deck 2:

- Three Slabs with Δ 's of 0.1, 0.4, and 0.5.
- One, Three, and One interior spatial edit points for the slabs.

```
3 16 3
11 41 2
10 1
0.0 1.0
1.0E-05
1.0 0.5
0.9 0.1 4 30 0.75 0.0 0.0 2
0.9 0.4 4 30 0.75 0.0 0.0 4
```

0.9 0.5 4 30 0.75 0.0 0.0 2

Input deck 3:

- Six Slabs with Δ 's of 0.05, 0.05, 0.1, 0.3, 0.25, and 0.25.
- No interior spatial edit points for any slab.

6 16 3
11 41 2
10 1
0.0 1.0
1.0E-05
1.0 0.5
0.9 0.05 4 30 0.75 0.0 0.0 1
0.9 0.05 4 30 0.75 0.0 0.0 1
0.9 0.1 4 30 0.75 0.0 0.0 1
0.9 0.3 4 30 0.75 0.0 0.0 1
0.9 0.25 4 30 0.75 0.0 0.0 1
0.9 0.25 4 30 0.75 0.0 0.0 1

V.C Output Description

The information concerning the program's output and where that output is to be found is contained in this section. It also contains the output generated by the example input decks of the preceding section.

The output files are as found in Table 6. The contents of the output files are found in Table 7. The screen output (also written to file number 26 named o6) shows the convergence criterion during the program's execution. The first column of screen output (NN =) contains the N of the F_N approximation. The second column (IE =) shows the number of boundary points that have converged. The third column (IEX =) shows the number of interior points that have converged. The final entries are the maximum number of boundary and interior points required to converge.

Table 6: List of Files Generated by MSLAB

File Number	File Name	File Contents
21	o1	Transmission and Reflection Coeff
22	o2	Boundary Fluxes For Each Slab
23	o3	Interior Edit (NXXR points) by Slab
26	o6	Screen Output

Table 7: Contents of Files Generated by MSLAB

File Number	File Name	File Contents with the Text Right of the Arrow Describing the Data in that Column
21	o1	Slab number → Reflected Coeff, Transmitted Coeff
22	o2	Mu vs Boundary Flux → abs(Mu), Flux for Mu<0, Flux for Mu>0
23	o3	Mu vs Interior Flux → abs(Mu), Flux for Mu<0, Flux for Mu>0

V.C.1 Sample Problem Output

The program's output due to the input decks from Section V.B.1 are shown below along with plotted results that match Figure 1 in Reference [1] (See Figure 1 in this manual).

Sample Problem 1

File o6:

NN=	11	IE=	0	IEX=	0	22	418
NN=	13	IE=	16	IEX=	0	22	418
NN=	15	IE=	18	IEX=	0	22	418
NN=	17	IE=	22	IEX=	0	22	418
NN=	19	IE=	22	IEX=	0	22	418
NN=	21	IE=	22	IEX=	404	22	418
NN=	23	IE=	22	IEX=	412	22	418
NN=	25	IE=	22	IEX=	416	22	418
NN=	27	IE=	22	IEX=	418	22	418

File o1:

SLAB	1
1.337052E-01	2.958126E-01
1.337052E-01	2.958125E-01
1.337052E-01	2.958125E-01
1.337052E-01	2.958125E-01
1.337052E-01	2.958125E-01
1.337052E-01	2.958125E-01
1.337052E-01	2.958125E-01
1.337052E-01	2.958125E-01
1.337052E-01	2.958125E-01

File o2:

FN SOLUTION
BDRY FLUXES

```

SLAB NR=      1 X=[ 0.0000000000000000E+00, 1.0000000000000000 ]
 1.00000E+00  2.10001E-01  1.90265E-01
 9.00000E-01  2.23885E-01  2.00704E-01
 8.00000E-01  2.39472E-01  2.11883E-01
 7.00000E-01  2.56974E-01  2.23633E-01
 6.00000E-01  2.76545E-01  2.35536E-01
 5.00000E-01  2.98151E-01  2.46679E-01
 4.00000E-01  3.21267E-01  2.55202E-01
 3.00000E-01  3.44241E-01  2.57582E-01
 2.00000E-01  3.63306E-01  2.48328E-01
 1.00000E-01  3.72669E-01  2.24033E-01
 1.00000E-10  3.59371E-01  1.86138E-01

```

File o3:

```

SLAB NR=      1 X=[ 0.0000000000000000E+00, 1.0000000000000000 ]
0.0000000000000000E+00  -
 1.00000E+00  2.10001E-01  0.00000E+00
 9.00000E-01  2.23885E-01  0.00000E+00
 8.00000E-01  2.39472E-01  0.00000E+00
 7.00000E-01  2.56974E-01  0.00000E+00
 6.00000E-01  2.76545E-01  0.00000E+00
 5.00000E-01  2.98151E-01  0.00000E+00
 4.00000E-01  3.21267E-01  0.00000E+00
 3.00000E-01  3.44241E-01  0.00000E+00
 2.00000E-01  3.63306E-01  0.00000E+00
 1.00000E-01  3.72669E-01  0.00000E+00
 1.00000E-10  3.59371E-01  0.00000E+00
5.0000000000000000E-02
 1.00000E+00  2.01846E-01  1.80042E-02
 9.00000E-01  2.15593E-01  1.99500E-02
 8.00000E-01  2.31116E-01  2.23673E-02
 7.00000E-01  2.48678E-01  2.54507E-02
 6.00000E-01  2.68510E-01  2.95197E-02
 5.00000E-01  2.90702E-01  3.51360E-02
 4.00000E-01  3.14917E-01  4.33880E-02
 3.00000E-01  3.39767E-01  5.66940E-02
 2.00000E-01  3.61744E-01  8.17099E-02
 1.00000E-01  3.75369E-01  1.45458E-01
 1.00000E-10  3.74858E-01  3.74858E-01
0.1000000000000000

```

1.00000E+00	1.92839E-01	3.55402E-02
9.00000E-01	2.06342E-01	3.92759E-02
8.00000E-01	2.21674E-01	4.38879E-02
7.00000E-01	2.39138E-01	4.97251E-02
6.00000E-01	2.59037E-01	5.73489E-02
5.00000E-01	2.81572E-01	6.77242E-02
4.00000E-01	3.06584E-01	8.26581E-02
3.00000E-01	3.32927E-01	1.05960E-01
2.00000E-01	3.57282E-01	1.47169E-01
1.00000E-01	3.74075E-01	2.36851E-01
1.00000E-10	3.79338E-01	3.79338E-01
0.1500000000000000		
1.00000E+00	1.83255E-01	5.23288E-02
9.00000E-01	1.96434E-01	5.76767E-02
8.00000E-01	2.11477E-01	6.42384E-02
7.00000E-01	2.28725E-01	7.24779E-02
6.00000E-01	2.48544E-01	8.31288E-02
5.00000E-01	2.71244E-01	9.74200E-02
4.00000E-01	2.96840E-01	1.17571E-01
3.00000E-01	3.24431E-01	1.47996E-01
2.00000E-01	3.50896E-01	1.98623E-01
1.00000E-01	3.70388E-01	2.93091E-01
1.00000E-10	3.79679E-01	3.79679E-01
0.2000000000000000		
1.00000E+00	1.73234E-01	6.82457E-02
9.00000E-01	1.86021E-01	7.50244E-02
8.00000E-01	2.00691E-01	8.32899E-02
7.00000E-01	2.17620E-01	9.35871E-02
6.00000E-01	2.37232E-01	1.06760E-01
5.00000E-01	2.59941E-01	1.24185E-01
4.00000E-01	2.85937E-01	1.48251E-01
3.00000E-01	3.14583E-01	1.83409E-01
2.00000E-01	3.42984E-01	2.38445E-01
1.00000E-01	3.64932E-01	3.26737E-01
1.00000E-10	3.77347E-01	3.77347E-01
0.2500000000000000		
1.00000E+00	1.62870E-01	8.32219E-02
9.00000E-01	1.75205E-01	9.12525E-02
8.00000E-01	1.89426E-01	1.00983E-01
7.00000E-01	2.05940E-01	1.13008E-01
6.00000E-01	2.25227E-01	1.28232E-01
5.00000E-01	2.47801E-01	1.48082E-01

4.00000E-01	2.74027E-01	1.74930E-01
3.00000E-01	3.03554E-01	2.12864E-01
2.00000E-01	3.33770E-01	2.68706E-01
1.00000E-01	3.58066E-01	3.45790E-01
1.00000E-10	3.73044E-01	3.73044E-01
0.3000000000000000		
1.00000E+00	1.52238E-01	9.72184E-02
9.00000E-01	1.64062E-01	1.06327E-01
8.00000E-01	1.77763E-01	1.17294E-01
7.00000E-01	1.93773E-01	1.30738E-01
6.00000E-01	2.12622E-01	1.47578E-01
5.00000E-01	2.34919E-01	1.69218E-01
4.00000E-01	2.61209E-01	1.97871E-01
3.00000E-01	2.91446E-01	2.37011E-01
2.00000E-01	3.23378E-01	2.91137E-01
1.00000E-01	3.50015E-01	3.55312E-01
1.00000E-10	3.67190E-01	3.67190E-01
0.3500000000000000		
1.00000E+00	1.41393E-01	1.10215E-01
9.00000E-01	1.52653E-01	1.20235E-01
8.00000E-01	1.65767E-01	1.32222E-01
7.00000E-01	1.81186E-01	1.46797E-01
6.00000E-01	1.99486E-01	1.64856E-01
5.00000E-01	2.21366E-01	1.87722E-01
4.00000E-01	2.47550E-01	2.17351E-01
3.00000E-01	2.78320E-01	2.56448E-01
2.00000E-01	3.11877E-01	3.07160E-01
1.00000E-01	3.40930E-01	3.58505E-01
1.00000E-10	3.60063E-01	3.60063E-01
0.4000000000000000		
1.00000E+00	1.30386E-01	1.22202E-01
9.00000E-01	1.41032E-01	1.32976E-01
8.00000E-01	1.53493E-01	1.45780E-01
7.00000E-01	1.68236E-01	1.61218E-01
6.00000E-01	1.85876E-01	1.80138E-01
5.00000E-01	2.07195E-01	2.03734E-01
4.00000E-01	2.33093E-01	2.33637E-01
3.00000E-01	2.64203E-01	2.71721E-01
2.00000E-01	2.99284E-01	3.17935E-01
1.00000E-01	3.30905E-01	3.57403E-01
1.00000E-10	3.51858E-01	3.51858E-01
0.4500000000000000		

1.00000E+00	1.19259E-01	1.33181E-01
9.00000E-01	1.29242E-01	1.44560E-01
8.00000E-01	1.40986E-01	1.57991E-01
7.00000E-01	1.54969E-01	1.74047E-01
6.00000E-01	1.71837E-01	1.93504E-01
5.00000E-01	1.92446E-01	2.17394E-01
4.00000E-01	2.17865E-01	2.46989E-01
3.00000E-01	2.49098E-01	2.83313E-01
2.00000E-01	2.85575E-01	3.24402E-01
1.00000E-01	3.19983E-01	3.53302E-01
1.00000E-10	3.42719E-01	3.42719E-01
0.5000000000000000		
1.00000E+00	1.08052E-01	1.43159E-01
9.00000E-01	1.17325E-01	1.55000E-01
8.00000E-01	1.28289E-01	1.68882E-01
7.00000E-01	1.41428E-01	1.85331E-01
6.00000E-01	1.57408E-01	2.05035E-01
5.00000E-01	1.77150E-01	2.28845E-01
4.00000E-01	2.01882E-01	2.57648E-01
3.00000E-01	2.32986E-01	2.91657E-01
2.00000E-01	2.70683E-01	3.27320E-01
1.00000E-01	3.08157E-01	3.47040E-01
1.00000E-10	3.32750E-01	3.32750E-01
0.5500000000000000		
1.00000E+00	9.68011E-02	1.52145E-01
9.00000E-01	1.05319E-01	1.64317E-01
8.00000E-01	1.15441E-01	1.78485E-01
7.00000E-01	1.27650E-01	1.95124E-01
6.00000E-01	1.42624E-01	2.14817E-01
5.00000E-01	1.61333E-01	2.38220E-01
4.00000E-01	1.85148E-01	2.65836E-01
3.00000E-01	2.15827E-01	2.97129E-01
2.00000E-01	2.54496E-01	3.27299E-01
1.00000E-01	2.95353E-01	3.39157E-01
1.00000E-10	3.22030E-01	3.22030E-01
0.6000000000000000		
1.00000E+00	8.55408E-02	1.60152E-01
9.00000E-01	9.32583E-02	1.72531E-01
8.00000E-01	1.02478E-01	1.86836E-01
7.00000E-01	1.13671E-01	2.03478E-01
6.00000E-01	1.27517E-01	2.22931E-01
5.00000E-01	1.45017E-01	2.45648E-01

4.00000E-01	1.67660E-01	2.71761E-01
3.00000E-01	1.97563E-01	3.00059E-01
2.00000E-01	2.36853E-01	3.24831E-01
1.00000E-01	2.81407E-01	3.30007E-01
1.00000E-10	3.10610E-01	3.10610E-01
0.6500000000000000		
1.00000E+00	7.43058E-02	1.67195E-01
9.00000E-01	8.11804E-02	1.79666E-01
8.00000E-01	8.94368E-02	1.93969E-01
7.00000E-01	9.95276E-02	2.10446E-01
6.00000E-01	1.12119E-01	2.29456E-01
5.00000E-01	1.28221E-01	2.51251E-01
4.00000E-01	1.49407E-01	2.75607E-01
3.00000E-01	1.78116E-01	3.00734E-01
2.00000E-01	2.17534E-01	3.20305E-01
1.00000E-01	2.66023E-01	3.19821E-01
1.00000E-10	2.98522E-01	2.98522E-01
0.7000000000000000		
1.00000E+00	6.31312E-02	1.73289E-01
9.00000E-01	6.91218E-02	1.85745E-01
8.00000E-01	7.63552E-02	1.99916E-01
7.00000E-01	8.52560E-02	2.16077E-01
6.00000E-01	9.64610E-02	2.34466E-01
5.00000E-01	1.10963E-01	2.55138E-01
4.00000E-01	1.30376E-01	2.77542E-01
3.00000E-01	1.57393E-01	2.99397E-01
2.00000E-01	1.96250E-01	3.14030E-01
1.00000E-01	2.48696E-01	3.08746E-01
1.00000E-10	2.85774E-01	2.85774E-01
0.7500000000000000		
1.00000E+00	5.20545E-02	1.78448E-01
9.00000E-01	5.71217E-02	1.90787E-01
8.00000E-01	6.32737E-02	2.04711E-01
7.00000E-01	7.08963E-02	2.20417E-01
6.00000E-01	8.05794E-02	2.38028E-01
5.00000E-01	9.32667E-02	2.57409E-01
4.00000E-01	1.10552E-01	2.77711E-01
3.00000E-01	1.35283E-01	2.96255E-01
2.00000E-01	1.72635E-01	3.06245E-01
1.00000E-01	2.28599E-01	2.96869E-01
1.00000E-10	2.72344E-01	2.72344E-01
0.8000000000000000		

1.00000E+00	4.11169E-02	1.82682E-01
9.00000E-01	4.52238E-02	1.94812E-01
8.00000E-01	5.02378E-02	2.08379E-01
7.00000E-01	5.64944E-02	2.23507E-01
6.00000E-01	6.45157E-02	2.40203E-01
5.00000E-01	7.51588E-02	2.58151E-01
4.00000E-01	8.99252E-02	2.76238E-01
3.00000E-01	1.11663E-01	2.91477E-01
2.00000E-01	1.46225E-01	2.97128E-01
1.00000E-01	2.04393E-01	2.84226E-01
1.00000E-10	2.58174E-01	2.58174E-01
0.8500000000000000		
1.00000E+00	3.03663E-02	1.85998E-01
9.00000E-01	3.34794E-02	1.97830E-01
8.00000E-01	3.73017E-02	2.10940E-01
7.00000E-01	4.21058E-02	2.25378E-01
6.00000E-01	4.83234E-02	2.41037E-01
5.00000E-01	5.66801E-02	2.57433E-01
4.00000E-01	6.84927E-02	2.73223E-01
3.00000E-01	8.64031E-02	2.85190E-01
2.00000E-01	1.16452E-01	2.86797E-01
1.00000E-01	1.73925E-01	2.70803E-01
1.00000E-10	2.43137E-01	2.43137E-01
0.9000000000000000		
1.00000E+00	1.98624E-02	1.88393E-01
9.00000E-01	2.19531E-02	1.99844E-01
8.00000E-01	2.45350E-02	2.12403E-01
7.00000E-01	2.78041E-02	2.26046E-01
6.00000E-01	3.20761E-02	2.40559E-01
5.00000E-01	3.78942E-02	2.55302E-01
4.00000E-01	4.62777E-02	2.68731E-01
3.00000E-01	5.93824E-02	2.77476E-01
2.00000E-01	8.26387E-02	2.75302E-01
1.00000E-01	1.33762E-01	2.56518E-01
1.00000E-10	2.26978E-01	2.26978E-01
0.9500000000000000		
1.00000E+00	9.68862E-03	1.89844E-01
9.00000E-01	1.07361E-02	2.00832E-01
8.00000E-01	1.20374E-02	2.12750E-01
7.00000E-01	1.36976E-02	2.25499E-01
6.00000E-01	1.58887E-02	2.38763E-01
5.00000E-01	1.89135E-02	2.51759E-01

4.00000E-01	2.33591E-02	2.62774E-01
3.00000E-01	3.05306E-02	2.68341E-01
2.00000E-01	4.40245E-02	2.62594E-01
1.00000E-01	7.84908E-02	2.41157E-01
1.00000E-10	2.09079E-01	2.09079E-01
1.0000000000000000		
1.00000E+00	0.00000E+00	1.90265E-01
9.00000E-01	0.00000E+00	2.00704E-01
8.00000E-01	0.00000E+00	2.11883E-01
7.00000E-01	0.00000E+00	2.23633E-01
6.00000E-01	0.00000E+00	2.35536E-01
5.00000E-01	0.00000E+00	2.46679E-01
4.00000E-01	0.00000E+00	2.55202E-01
3.00000E-01	0.00000E+00	2.57582E-01
2.00000E-01	0.00000E+00	2.48328E-01
1.00000E-01	0.00000E+00	2.24033E-01
1.00000E-10	0.00000E+00	1.86138E-01

Sample Problem 2

File o6:

NN=	11	IE=	0	IEX=	0	66	110
NN=	11	IE=	0	IEX=	0	66	110
NN=	11	IE=	0	IEX=	0	66	110
NN=	13	IE=	0	IEX=	0	66	110
NN=	13	IE=	0	IEX=	0	66	110
NN=	13	IE=	0	IEX=	0	66	110
NN=	15	IE=	14	IEX=	0	66	110
NN=	15	IE=	15	IEX=	0	66	110
NN=	15	IE=	15	IEX=	0	66	110
NN=	17	IE=	51	IEX=	0	66	110
NN=	17	IE=	50	IEX=	0	66	110
NN=	17	IE=	50	IEX=	0	66	110
NN=	19	IE=	59	IEX=	0	66	110
NN=	19	IE=	59	IEX=	0	66	110
NN=	19	IE=	59	IEX=	0	66	110
NN=	21	IE=	63	IEX=	0	66	110
NN=	21	IE=	63	IEX=	0	66	110
NN=	21	IE=	63	IEX=	0	66	110
NN=	23	IE=	66	IEX=	0	66	110
NN=	23	IE=	66	IEX=	0	66	110
NN=	25	IE=	66	IEX=	109	66	110
NN=	27	IE=	66	IEX=	110	66	110

File o1:

SLAB	1
1.285502E-01	4.832700E-01
1.336903E-01	4.837045E-01
1.337049E-01	4.837057E-01
1.337051E-01	4.837056E-01
1.337052E-01	4.837056E-01
1.337052E-01	4.837056E-01
1.337052E-01	4.837056E-01
1.337052E-01	4.837056E-01
1.337052E-01	4.837056E-01

SLAB	2
1.248301E-01	4.033682E-01
1.257041E-01	4.039195E-01
1.257066E-01	4.039214E-01
1.257069E-01	4.039218E-01
1.257070E-01	4.039219E-01
1.257070E-01	4.039219E-01
1.257070E-01	4.039219E-01
1.257069E-01	4.039218E-01
1.257069E-01	4.039218E-01

SLAB	3
7.809918E-02	2.954598E-01
7.824439E-02	2.958111E-01
7.824487E-02	2.958122E-01
7.824500E-02	2.958125E-01
7.824504E-02	2.958126E-01
7.824503E-02	2.958126E-01
7.824503E-02	2.958126E-01
7.824502E-02	2.958126E-01
7.824502E-02	2.958126E-01

File o2:

FN SOLUTION
BDRY FLUXES

SLAB NR=	1	X=[	0.0000000000000000E+00,	0.1000000000000000	]
1.00000E+00	2.10001E-01	3.55403E-02			
9.00000E-01	2.23885E-01	3.92760E-02			
8.00000E-01	2.39472E-01	4.38880E-02			
7.00000E-01	2.56974E-01	4.97252E-02			
6.00000E-01	2.76545E-01	5.73490E-02			
5.00000E-01	2.98152E-01	6.77244E-02			
4.00000E-01	3.21267E-01	8.26583E-02			
3.00000E-01	3.44241E-01	1.05961E-01			
2.00000E-01	3.63307E-01	1.47170E-01			
1.00000E-01	3.72670E-01	2.36851E-01			
1.00000E-10	3.59367E-01	3.79340E-01			
SLAB NR=	2	X=[	0.1000000000000000	, 0.5000000000000000	]
1.00000E+00	1.92839E-01	1.43159E-01			

9.00000E-01	2.06342E-01	1.55000E-01	
8.00000E-01	2.21674E-01	1.68882E-01	
7.00000E-01	2.39138E-01	1.85332E-01	
6.00000E-01	2.59037E-01	2.05036E-01	
5.00000E-01	2.81572E-01	2.28845E-01	
4.00000E-01	3.06585E-01	2.57648E-01	
3.00000E-01	3.32927E-01	2.91657E-01	
2.00000E-01	3.57282E-01	3.27320E-01	
1.00000E-01	3.74076E-01	3.47040E-01	
1.00000E-10	3.79344E-01	3.32751E-01	
SLAB NR=	3 X=[	0.5000000000000000	, 1.0000000000000000]
1.00000E+00	1.08052E-01	1.90265E-01	
9.00000E-01	1.17325E-01	2.00704E-01	
8.00000E-01	1.28289E-01	2.11883E-01	
7.00000E-01	1.41428E-01	2.23634E-01	
6.00000E-01	1.57408E-01	2.35536E-01	
5.00000E-01	1.77150E-01	2.46679E-01	
4.00000E-01	2.01882E-01	2.55202E-01	
3.00000E-01	2.32986E-01	2.57582E-01	
2.00000E-01	2.70683E-01	2.48328E-01	
1.00000E-01	3.08157E-01	2.24033E-01	
1.00000E-10	3.32751E-01	1.86138E-01	

File 03:

SLAB NR=	1 X=[	0.0000000000000000E+00,	0.1000000000000000	]
0.0000000000000000E+00				
1.00000E+00	2.10001E-01	0.00000E+00		
9.00000E-01	2.23885E-01	0.00000E+00		
8.00000E-01	2.39472E-01	0.00000E+00		
7.00000E-01	2.56974E-01	0.00000E+00		
6.00000E-01	2.76545E-01	0.00000E+00		
5.00000E-01	2.98152E-01	0.00000E+00		
4.00000E-01	3.21267E-01	0.00000E+00		
3.00000E-01	3.44241E-01	0.00000E+00		
2.00000E-01	3.63307E-01	0.00000E+00		
1.00000E-01	3.72670E-01	0.00000E+00		
1.00000E-10	3.59367E-01	0.00000E+00		
5.0000000000000000E-02				
1.00000E+00	2.01847E-01	1.80042E-02		
9.00000E-01	2.15593E-01	1.99500E-02		

8.00000E-01	2.31116E-01	2.23673E-02	
7.00000E-01	2.48678E-01	2.54507E-02	
6.00000E-01	2.68510E-01	2.95197E-02	
5.00000E-01	2.90702E-01	3.51360E-02	
4.00000E-01	3.14918E-01	4.33880E-02	
3.00000E-01	3.39768E-01	5.66940E-02	
2.00000E-01	3.61745E-01	8.17100E-02	
1.00000E-01	3.75369E-01	1.45458E-01	
1.00000E-10	3.74876E-01	3.74876E-01	
SLAB NR=	2 X=[	0.1000000000000000	, 0.5000000000000000]
0.1000000000000000			
1.00000E+00	1.92839E-01	3.55403E-02	
9.00000E-01	2.06342E-01	3.92760E-02	
8.00000E-01	2.21674E-01	4.38880E-02	
7.00000E-01	2.39138E-01	4.97252E-02	
6.00000E-01	2.59037E-01	5.73490E-02	
5.00000E-01	2.81572E-01	6.77243E-02	
4.00000E-01	3.06585E-01	8.26583E-02	
3.00000E-01	3.32927E-01	1.05960E-01	
2.00000E-01	3.57282E-01	1.47170E-01	
1.00000E-01	3.74076E-01	2.36851E-01	
1.00000E-10	3.79344E-01	3.79339E-01	
0.2000000000000000			
1.00000E+00	1.73234E-01	6.82458E-02	
9.00000E-01	1.86021E-01	7.50246E-02	
8.00000E-01	2.00691E-01	8.32901E-02	
7.00000E-01	2.17620E-01	9.35873E-02	
6.00000E-01	2.37232E-01	1.06760E-01	
5.00000E-01	2.59941E-01	1.24186E-01	
4.00000E-01	2.85937E-01	1.48252E-01	
3.00000E-01	3.14583E-01	1.83409E-01	
2.00000E-01	3.42984E-01	2.38446E-01	
1.00000E-01	3.64932E-01	3.26739E-01	
1.00000E-10	3.77340E-01	3.77340E-01	
0.3000000000000000			
1.00000E+00	1.52238E-01	9.72185E-02	
9.00000E-01	1.64062E-01	1.06327E-01	
8.00000E-01	1.77763E-01	1.17294E-01	
7.00000E-01	1.93773E-01	1.30739E-01	
6.00000E-01	2.12622E-01	1.47578E-01	
5.00000E-01	2.34919E-01	1.69218E-01	
4.00000E-01	2.61209E-01	1.97871E-01	

3.00000E-01	2.91446E-01	2.37011E-01	
2.00000E-01	3.23379E-01	2.91137E-01	
1.00000E-01	3.50015E-01	3.55312E-01	
1.00000E-10	3.67192E-01	3.67192E-01	
0.4000000000000000			
1.00000E+00	1.30386E-01	1.22202E-01	
9.00000E-01	1.41032E-01	1.32977E-01	
8.00000E-01	1.53493E-01	1.45780E-01	
7.00000E-01	1.68236E-01	1.61218E-01	
6.00000E-01	1.85876E-01	1.80138E-01	
5.00000E-01	2.07195E-01	2.03734E-01	
4.00000E-01	2.33093E-01	2.33638E-01	
3.00000E-01	2.64203E-01	2.71721E-01	
2.00000E-01	2.99284E-01	3.17935E-01	
1.00000E-01	3.30906E-01	3.57402E-01	
1.00000E-10	3.51854E-01	3.51854E-01	
SLAB NR=	3 X=[0.5000000000000000	, 1.0000000000000000	]
0.5000000000000000			
1.00000E+00	1.08052E-01	1.43159E-01	
9.00000E-01	1.17325E-01	1.55000E-01	
8.00000E-01	1.28289E-01	1.68882E-01	
7.00000E-01	1.41428E-01	1.85332E-01	
6.00000E-01	1.57408E-01	2.05035E-01	
5.00000E-01	1.77150E-01	2.28845E-01	
4.00000E-01	2.01882E-01	2.57648E-01	
3.00000E-01	2.32986E-01	2.91657E-01	
2.00000E-01	2.70683E-01	3.27320E-01	
1.00000E-01	3.08157E-01	3.47040E-01	
1.00000E-10	3.32751E-01	3.32751E-01	
0.7500000000000000			
1.00000E+00	5.20545E-02	1.78448E-01	
9.00000E-01	5.71217E-02	1.90787E-01	
8.00000E-01	6.32737E-02	2.04711E-01	
7.00000E-01	7.08963E-02	2.20417E-01	
6.00000E-01	8.05794E-02	2.38029E-01	
5.00000E-01	9.32667E-02	2.57409E-01	
4.00000E-01	1.10552E-01	2.77711E-01	
3.00000E-01	1.35283E-01	2.96255E-01	
2.00000E-01	1.72635E-01	3.06246E-01	
1.00000E-01	2.28599E-01	2.96869E-01	
1.00000E-10	2.72345E-01	2.72345E-01	
1.0000000000000000			

1.00000E+00	0.00000E+00	1.90265E-01
9.00000E-01	0.00000E+00	2.00704E-01
8.00000E-01	0.00000E+00	2.11883E-01
7.00000E-01	0.00000E+00	2.23634E-01
6.00000E-01	0.00000E+00	2.35536E-01
5.00000E-01	0.00000E+00	2.46679E-01
4.00000E-01	0.00000E+00	2.55202E-01
3.00000E-01	0.00000E+00	2.57582E-01
2.00000E-01	0.00000E+00	2.48328E-01
1.00000E-01	0.00000E+00	2.24033E-01
1.00000E-10	0.00000E+00	1.86138E-01

Sample Problem 3

File o6:

NN=	11 IE=	0 IEX=	0	132	0
NN=	11 IE=	0 IEX=	0	132	0
NN=	11 IE=	0 IEX=	0	132	0
NN=	13 IE=	0 IEX=	0	132	0
NN=	13 IE=	0 IEX=	0	132	0
NN=	13 IE=	0 IEX=	0	132	0
NN=	15 IE=	0 IEX=	0	132	0
NN=	15 IE=	0 IEX=	0	132	0
NN=	15 IE=	0 IEX=	0	132	0
NN=	17 IE=	0 IEX=	0	132	0
NN=	17 IE=	0 IEX=	0	132	0
NN=	17 IE=	0 IEX=	0	132	0
NN=	19 IE=	58 IEX=	0	132	0
NN=	19 IE=	47 IEX=	0	132	0
NN=	19 IE=	40 IEX=	0	132	0
NN=	21 IE=	117 IEX=	0	132	0
NN=	21 IE=	108 IEX=	0	132	0
NN=	21 IE=	103 IEX=	0	132	0
NN=	23 IE=	122 IEX=	0	132	0
NN=	23 IE=	121 IEX=	0	132	0
NN=	23 IE=	122 IEX=	0	132	0
NN=	25 IE=	130 IEX=	0	132	0
NN=	25 IE=	129 IEX=	0	132	0
NN=	25 IE=	129 IEX=	0	132	0
NN=	27 IE=	132 IEX=	0	132	0
NN=	27 IE=	132 IEX=	0	132	0

File o1:

SLAB	1
3.937156E-02	4.881898E-01
1.281899E-01	4.920301E-01
1.333497E-01	4.922490E-01
1.336872E-01	4.922631E-01
1.337041E-01	4.922638E-01

1.337052E-01	4.922638E-01
1.337054E-01	4.922638E-01
1.337055E-01	4.922638E-01
1.337056E-01	4.922638E-01

SLAB	2
7.580211E-02	4.777190E-01
1.276844E-01	4.833899E-01
1.299296E-01	4.836864E-01
1.300631E-01	4.837045E-01
1.300698E-01	4.837054E-01
1.300703E-01	4.837056E-01
1.300704E-01	4.837057E-01
1.300704E-01	4.837058E-01
1.300704E-01	4.837058E-01

SLAB	3
9.772624E-02	4.576990E-01
1.246937E-01	4.647904E-01
1.256555E-01	4.651281E-01
1.257041E-01	4.651474E-01
1.257067E-01	4.651484E-01
1.257069E-01	4.651487E-01
1.257070E-01	4.651489E-01
1.257071E-01	4.651490E-01
1.257071E-01	4.651491E-01

SLAB	4
1.092035E-01	3.972311E-01
1.151586E-01	4.035690E-01
1.155566E-01	4.039019E-01
1.155764E-01	4.039205E-01
1.155775E-01	4.039216E-01
1.155776E-01	4.039218E-01
1.155777E-01	4.039220E-01
1.155777E-01	4.039220E-01
1.155777E-01	4.039220E-01

SLAB	5
7.555241E-02	3.448500E-01
7.807422E-02	3.501960E-01
7.823655E-02	3.504831E-01

7.824449E-02	3.504989E-01
7.824496E-02	3.504998E-01
7.824505E-02	3.505001E-01
7.824509E-02	3.505002E-01
7.824510E-02	3.505002E-01
7.824510E-02	3.505002E-01

SLAB	6
4.083015E-02	2.913136E-01
4.166200E-02	2.955713E-01
4.170692E-02	2.957992E-01
4.170934E-02	2.958117E-01
4.170950E-02	2.958124E-01
4.170954E-02	2.958126E-01
4.170955E-02	2.958127E-01
4.170956E-02	2.958128E-01
4.170956E-02	2.958128E-01

File o2:

FN SOLUTION

BDRY FLUXES

SLAB NR= 1 X=[0.0000000000000000E+00, 5.0000000000000000E-02]

1.00000E+00	2.10001E-01	1.80043E-02
9.00000E-01	2.23886E-01	1.99501E-02
8.00000E-01	2.39473E-01	2.23673E-02
7.00000E-01	2.56974E-01	2.54508E-02
6.00000E-01	2.76546E-01	2.95198E-02
5.00000E-01	2.98152E-01	3.51361E-02
4.00000E-01	3.21268E-01	4.33882E-02
3.00000E-01	3.44242E-01	5.66942E-02
2.00000E-01	3.63308E-01	8.17101E-02
1.00000E-01	3.72672E-01	1.45459E-01
1.00000E-10	3.59389E-01	3.74860E-01

SLAB NR= 2 X=[5.0000000000000000E-02, 0.1000000000000000]

1.00000E+00	2.01847E-01	3.55405E-02
9.00000E-01	2.15593E-01	3.92761E-02
8.00000E-01	2.31117E-01	4.38882E-02
7.00000E-01	2.48678E-01	4.97254E-02
6.00000E-01	2.68511E-01	5.73492E-02
5.00000E-01	2.90703E-01	6.77247E-02

4.00000E-01	3.14918E-01	8.26587E-02	
3.00000E-01	3.39769E-01	1.05961E-01	
2.00000E-01	3.61746E-01	1.47170E-01	
1.00000E-01	3.75371E-01	2.36852E-01	
1.00000E-10	3.74861E-01	3.79362E-01	
SLAB NR=	3 X=[	0.1000000000000000	, 0.2000000000000000]
1.00000E+00	1.92840E-01	6.82461E-02	
9.00000E-01	2.06342E-01	7.50249E-02	
8.00000E-01	2.21674E-01	8.32905E-02	
7.00000E-01	2.39138E-01	9.35877E-02	
6.00000E-01	2.59037E-01	1.06761E-01	
5.00000E-01	2.81573E-01	1.24186E-01	
4.00000E-01	3.06585E-01	1.48252E-01	
3.00000E-01	3.32928E-01	1.83410E-01	
2.00000E-01	3.57283E-01	2.38446E-01	
1.00000E-01	3.74077E-01	3.26739E-01	
1.00000E-10	3.79340E-01	3.77342E-01	
SLAB NR=	4 X=[	0.2000000000000000	, 0.5000000000000000]
1.00000E+00	1.73234E-01	1.43159E-01	
9.00000E-01	1.86021E-01	1.55001E-01	
8.00000E-01	2.00692E-01	1.68882E-01	
7.00000E-01	2.17620E-01	1.85332E-01	
6.00000E-01	2.37232E-01	2.05036E-01	
5.00000E-01	2.59941E-01	2.28845E-01	
4.00000E-01	2.85937E-01	2.57648E-01	
3.00000E-01	3.14584E-01	2.91658E-01	
2.00000E-01	3.42985E-01	3.27321E-01	
1.00000E-01	3.64933E-01	3.47041E-01	
1.00000E-10	3.77347E-01	3.32752E-01	
SLAB NR=	5 X=[	0.5000000000000000	, 0.7500000000000000]
1.00000E+00	1.08052E-01	1.78448E-01	
9.00000E-01	1.17326E-01	1.90788E-01	
8.00000E-01	1.28290E-01	2.04711E-01	
7.00000E-01	1.41428E-01	2.20418E-01	
6.00000E-01	1.57408E-01	2.38029E-01	
5.00000E-01	1.77150E-01	2.57410E-01	
4.00000E-01	2.01883E-01	2.77711E-01	
3.00000E-01	2.32986E-01	2.96256E-01	
2.00000E-01	2.70683E-01	3.06246E-01	
1.00000E-01	3.08158E-01	2.96869E-01	
1.00000E-10	3.32752E-01	2.72345E-01	
SLAB NR=	6 X=[	0.7500000000000000	, 1.0000000000000000]

1.00000E+00	5.20546E-02	1.90265E-01
9.00000E-01	5.71218E-02	2.00705E-01
8.00000E-01	6.32738E-02	2.11884E-01
7.00000E-01	7.08964E-02	2.23634E-01
6.00000E-01	8.05795E-02	2.35537E-01
5.00000E-01	9.32668E-02	2.46679E-01
4.00000E-01	1.10553E-01	2.55203E-01
3.00000E-01	1.35283E-01	2.57582E-01
2.00000E-01	1.72635E-01	2.48328E-01
1.00000E-01	2.28599E-01	2.24033E-01
1.00000E-10	2.72345E-01	1.86138E-01

File o3:

SLAB NR= 1 X=[0.0000000000000000E+00, 5.0000000000000000E-02]
0.0000000000000000E+00

1.00000E+00	2.10001E-01	0.00000E+00
9.00000E-01	2.23886E-01	0.00000E+00
8.00000E-01	2.39473E-01	0.00000E+00
7.00000E-01	2.56974E-01	0.00000E+00
6.00000E-01	2.76546E-01	0.00000E+00
5.00000E-01	2.98152E-01	0.00000E+00
4.00000E-01	3.21268E-01	0.00000E+00
3.00000E-01	3.44242E-01	0.00000E+00
2.00000E-01	3.63308E-01	0.00000E+00
1.00000E-01	3.72672E-01	0.00000E+00
1.00000E-10	3.59389E-01	0.00000E+00

SLAB NR= 2 X=[5.0000000000000000E-02, 0.1000000000000000]
5.0000000000000000E-02

1.00000E+00	2.01847E-01	1.80043E-02
9.00000E-01	2.15593E-01	1.99501E-02
8.00000E-01	2.31117E-01	2.23674E-02
7.00000E-01	2.48678E-01	2.54508E-02
6.00000E-01	2.68511E-01	2.95198E-02
5.00000E-01	2.90703E-01	3.51361E-02
4.00000E-01	3.14918E-01	4.33882E-02
3.00000E-01	3.39769E-01	5.66943E-02
2.00000E-01	3.61746E-01	8.17103E-02
1.00000E-01	3.75371E-01	1.45459E-01
1.00000E-10	3.74861E-01	3.74860E-01

SLAB NR= 3 X=[0.1000000000000000 , 0.2000000000000000]

0.10000000000000000

1.00000E+00	1.92840E-01	3.55405E-02
9.00000E-01	2.06342E-01	3.92762E-02
8.00000E-01	2.21674E-01	4.38883E-02
7.00000E-01	2.39138E-01	4.97255E-02
6.00000E-01	2.59037E-01	5.73493E-02
5.00000E-01	2.81573E-01	6.77247E-02
4.00000E-01	3.06585E-01	8.26587E-02
3.00000E-01	3.32928E-01	1.05961E-01
2.00000E-01	3.57283E-01	1.47170E-01
1.00000E-01	3.74077E-01	2.36851E-01
1.00000E-10	3.79340E-01	3.79361E-01

SLAB NR= 4 X=[0.20000000000000000

, 0.50000000000000000]

0.20000000000000000

1.00000E+00	1.73234E-01	6.82461E-02
9.00000E-01	1.86021E-01	7.50249E-02
8.00000E-01	2.00692E-01	8.32905E-02
7.00000E-01	2.17620E-01	9.35877E-02
6.00000E-01	2.37232E-01	1.06761E-01
5.00000E-01	2.59941E-01	1.24186E-01
4.00000E-01	2.85937E-01	1.48252E-01
3.00000E-01	3.14584E-01	1.83410E-01
2.00000E-01	3.42985E-01	2.38446E-01
1.00000E-01	3.64933E-01	3.26739E-01
1.00000E-10	3.77347E-01	3.77341E-01

SLAB NR= 5 X=[0.50000000000000000

, 0.75000000000000000]

0.50000000000000000

1.00000E+00	1.08052E-01	1.43159E-01
9.00000E-01	1.17326E-01	1.55001E-01
8.00000E-01	1.28290E-01	1.68882E-01
7.00000E-01	1.41428E-01	1.85332E-01
6.00000E-01	1.57408E-01	2.05036E-01
5.00000E-01	1.77150E-01	2.28845E-01
4.00000E-01	2.01883E-01	2.57648E-01
3.00000E-01	2.32986E-01	2.91658E-01
2.00000E-01	2.70683E-01	3.27321E-01
1.00000E-01	3.08158E-01	3.47041E-01
1.00000E-10	3.32752E-01	3.32751E-01

SLAB NR= 6 X=[0.75000000000000000

, 1.00000000000000000]

0.75000000000000000

1.00000E+00	5.20546E-02	1.78448E-01
9.00000E-01	5.71218E-02	1.90788E-01

8.00000E-01	6.32738E-02	2.04711E-01
7.00000E-01	7.08964E-02	2.20418E-01
6.00000E-01	8.05795E-02	2.38029E-01
5.00000E-01	9.32668E-02	2.57410E-01
4.00000E-01	1.10553E-01	2.77711E-01
3.00000E-01	1.35283E-01	2.96256E-01
2.00000E-01	1.72635E-01	3.06246E-01
1.00000E-01	2.28599E-01	2.96869E-01
1.00000E-10	2.72345E-01	2.72345E-01
1.0000000000000000		
1.00000E+00	0.00000E+00	1.90265E-01
9.00000E-01	0.00000E+00	2.00705E-01
8.00000E-01	0.00000E+00	2.11884E-01
7.00000E-01	0.00000E+00	2.23634E-01
6.00000E-01	0.00000E+00	2.35537E-01
5.00000E-01	0.00000E+00	2.46679E-01
4.00000E-01	0.00000E+00	2.55203E-01
3.00000E-01	0.00000E+00	2.57582E-01
2.00000E-01	0.00000E+00	2.48328E-01
1.00000E-01	0.00000E+00	2.24033E-01
1.00000E-10	0.00000E+00	1.86138E-01

Flux versus Direction for MultiSlab FN Method

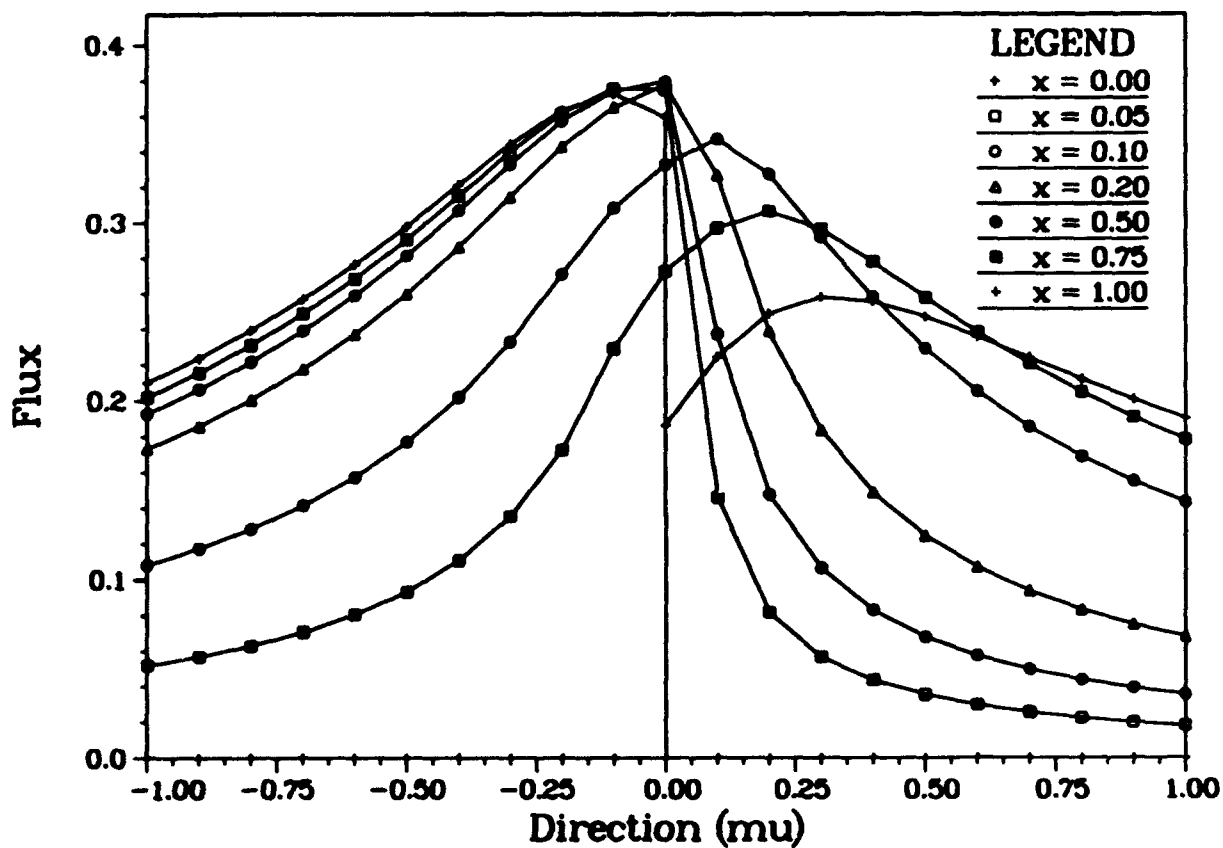


Figure 1: Regeneration of Siewert's Data from Sample Problem Data

V.D Detailed Program Notes

Detailed programming notes are given in this section. These notes refer to comments and detailed instructions deemed necessary for the programmer who would like to understand and modify the MSLAB code.

V.D.1 Main Program

These are the comments for the Main Program.

1. CMAIN1 *** READ INPUT

- (a) The following are the current limits contained in PARAMETER statements at the beginning of the main program and where required in subroutines:

NP = Maximum number of N of F_N approximation (70).

ILM = Maximum order of Gauss-Legendre (G-L) quadrature (100).

IMM1 = Maximum number of angular edit points (100).

INS = Maximum number of slabs (10).

NPX = Maximum number of spatial edit points within each slab (50).

The program dimensions are automatically adjusted by changing the values of the above variables in the PARAMETER statements.

- (b) If only one slab is being considered ($NS = 1$), the number of boundary iterations (LIT) is set to 1.
- (c) The G-L quadrature order required to determine the boundary or interior fluxes can be different for each slab but the G-L quadrature order for the evaluation of integrals over the boundary fluxes connecting the slabs will always be the same for each slab.
- (d) Four possibilities for basis functions have been provided. Table 8 lists these functions. From numerical experiment, the value of γ (XAR) = 0.75 (for IBR = 3 or 4) has been found to give the best results.

2. CMAIN5 *** F_N ITERATION LOOP

- (a) This is the main iteration loop which cycles through the F_N approximation for increasing N.
- (b) N is incremented by L12D specified on Line 2 of the input.

3. CMAIN7 *** BOUNDARY FLUX ITERATION LOOP

Since the boundary fluxes at the slab interfaces for the multiple slab case are not known *a priori*, an iterative process is required for their determination. By numerical experiment, it has been verified that rather than having the F_N iteration serve also as a boundary flux iteration, it is more efficient to include at least several boundary flux iterations at each N . This reduces the N required for convergence. However, there is a tradeoff since if many boundary flux iterations are included, the computational time will be greatly increased. In addition the advantage this iteration provides is reduced as N increases.

4. CMAIN8 *** SLAB LOOP

This loop cycles through the slabs with the previously determined boundary fluxes.

5. CMAIN10 *** CHECK TO AVOID SAME QUADRATURE ABSCISSA EDIT POINTS AND COLLOCATION POINTS

Since 0.5 is a zero of all odd order shifted Legendre polynomials, a modification must be made to avoid division by zero in some parts of the calculation. This is accomplished by not allowing the F_N order N (NN) and the quadrature order for the determination of the matrix elements (LMR) to both be odd (or even) simultaneously. If they are both odd or even, then LMR is increased by one. A similar procedure is required for the quadrature order for the integration of the boundary fluxes (LM3).

6. CMAIN13 *** SPECIFY END OR INTERIOR SLAB AND BOUNDARY CONDITIONS (BC)

In subroutine S1, a slab interface indicator for the right (IBDR) and left (IBDL) interfaces is set according to the values in Table 9. These flags are necessary for the appropriate integration of the boundary fluxes (subroutines INTO, INTR, INTRZ) to form the inhomogeneous terms of the collocation equations.

7. CMAIN16 *** FIND MATRIX ELEMENTS AND INHOMOGENEOUS TERMS

When IFLG = 1, the matrix elements and inhomogeneous terms are formed in the symmetric form in subroutine MELB to be used in the boundary flux determination [eqs. (25) and eqs. (26)]. For IFLG = -1, the matrix elements and inhomogeneous terms are put in the asymmetric form for eqs. (20).

8. CMAIN19 *** CHECK F_N CONVERGENCE FOR BOUNDARY FLUXES AT THE EDIT POINTS

Before the calculation of the interior flux is initiated, the boundary fluxes must have converged relative to the last F_N determination at the desired angular edit point. Once the boundary fluxes have converged, then the interior fluxes are determined until they converge at all angular and spatial edit points. Even though the boundary fluxes have converged, they are recalculated at the same F_N approximation for the interior fluxes since they are required at the same collocation points.

Table 8: Basis Functions Available to Program and How to Access Them

IB	Name	$\psi_{\alpha}(\mu)$
1	Shifted Legendre polynomials	$P_{\alpha}(2\mu - 1)$
2	Monomials	μ^{α}
3	Shifted Monomials	$(2\mu^{\gamma} - 1)^{\alpha}$
4	Modified Legendre polynomials	$P_{\alpha}(2\mu^{\gamma} - 1)$

Table 9: Interface/Boundary Condition and How to Access Them

IBDR or IBDL	Interface/Boundary Condition
0	Slab/Slab no source
1	Vacuum/Slab with beam source
2	Slab/Vacuum no source

V.D.2 FNX Subprogram

These are the comments for the FNX Subprogram. Its main purpose is to calculate the values of the flux in the interior of a slab.

1. CFNX1 *** INITIALIZE CONVERGENCE FLAG IPS
CFNX1 *** IPS(IR+NR,IX) = -1 POINT IX IN SLAB NR NOT YET CONVERGED
CFNX1 *** IPS(IR+NR,IX) = 1 POINT IX IN SLAB NR HAS CONVERGED
Initialize all convergence pointers to the *not converged* value (-1).
2. CFNX2 *** DETERMINE MATRIX ELEMENTS FOR CURRENT COLLOCATION POINTS AND REQUIRED SUMS
When IFLG = -2, the matrix elements are calculated in subprogram MELB. The A and B function values are found and summed.
3. CFNX3 *** SPATIAL GRID LOOP
If the point has already converged, then it is skipped. If not, then determine where it is in the slab and set up some flags for the next step.
4. CFNX4 *** SET UP MATRIX ELEMENTS AND INHOMOGENEOUS TERM FOR INTERIOR FLUX CALCULATION
For IFLG = -1, the matrix elements and inhomogeneous terms are put in the asymmetric form of eqs. (20) and calculated using the MELB subprogram.
5. CFNX5 *** SOLVE FOR THE COUPLING COEFFICIENTS
Performs the matrix inversion required to solve for the coupling coefficients in eqs. (20).
6. CFNX6 *** DETERMINE INTERIOR FLUXES
Using the post-processing algorithm, the interior fluxes are found using eqs. (21).
7. CFNX7 *** TEST FOR CONVERGENCE
If the relative error from the last iteration is below the tolerance, then set the converged flag. If not, then continue.

VI Acknowledgement

I acknowledge the generous help and encouragement given to me in the development of the work by J. C. Garth (at RADC) who made it all possible. To C. E. Siewert, who originally developed the F_N method, I say thanks for your insight and friendship. Finally, to R. C. Singleterry, goes my heartfelt thanks for a beautiful typing and editing job.

References

- [1] R. C. Y. Chin, G. W. Hedstrom, and C. E. Siewert, *On the Use of the F_N Method with Splines for Radiative Transfer Problems*, UCRL-94464, April 17, 1986.
- [2] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes, The Art of Scientific Computing*, Cambridge University Press, 1986.

4

Multigroup- S_n Benchmark Comparison for Electron Transport

B. D. Ganapol
University of Arizona, Department of Nuclear and Energy Engineering
Tucson, AZ 85721

S. Woolf
ARCON Corporation
Waltham, MA 02154

J. C. Garth
Rome Air Development Center, Solid State Sciences Division
Hanscom AFB, MA 01731

ABSTRACT

A set of benchmark calculations is described for the one-dimensional, multigroup discrete ordinates method as applied to electron transport. An analytical representation of the group scalar flux is obtained by application of the Fourier transform operator to the transport equation, expansion of the transformed equation in terms of the flux moments, and re-inversion of the transformed scalar flux moments. The results of S_n calculations, with both the linear discontinuous and diamond difference algorithms, are compared with benchmark calculations for a plane isotropic source of 200 keV electrons in infinite aluminum. The degree of scattering anisotropy is allowed to vary from isotropic to third-order anisotropic. Analysis of spatial discretization error is also discussed.

I. Introduction

An integral part of transport method development is verification that the numerical algorithm used, as well as its coding, perform properly. Unfortunately, due to the lack of appropriate benchmarks and the fact that verification efforts are usually tangential to the main thrust of the development, code verification is often not seriously considered. Of course, a full verification is not possible since if it were, there would be no need for the method development in the first place. Partial verification however, can be achieved by application of any or all of the following procedures:

- a) monitoring of particle conservation;
- b) comparison with intuitive trends;
- c) comparison to similar algorithms;
- d) comparison to analytical solutions.

It should be emphasized that the success of a verification procedure does not provide proof positive that the method is operational. However, the lack thereof does indicate that the numerical algorithm or its coding is in some way defective. This presentation is concerned with the establishment of a benchmark for 1-D multigroup transport algorithms and the verification of a specific S_n algorithm used in electron transport theory, as an example.

II. Multigroup Benchmark Formulation

The slowing down and deflection of electrons in a medium larger than the electron range (effectively a planar infinite medium) can be approximated by the following multigroup formulation, where all the symbols have their usual meaning.^{1,2}

30.1 9-1

$$\left(\mu \frac{\partial}{\partial x} + \frac{1}{\lambda_g}\right) \psi_g(x, \mu) = \sum_{g'=1}^g \sum_{l=0}^L \frac{2l+1}{2} \Sigma_{l,g',g} P_l(\mu) \int_{-1}^1 d\mu' P_l(\mu') \psi_{g'}(x, \mu') + Q(\mu) \delta(x) \delta_{g,1} \quad (1)$$

The group mean free path λ_g and group transfer cross sections $\Sigma_{l,g',g}$ are determined assuming that continuous slowing down theory is applicable¹. A source has been placed at the center of the infinite medium and is assumed to emit electrons in the highest energy group with a beam or isotropic angular distribution

$$Q(\mu) = \begin{cases} \delta(\mu - \mu_0), & \text{beam} \\ 1/2, & \text{isotropic} \end{cases}$$

The analytical representation of the group scalar flux $\psi_g(x)$ is obtained by application of the Fourier transform operator to Eq. (1) in the form

$$\bar{\psi}_g(k, \mu) \equiv \int_{-\infty}^{\infty} dx e^{-ikx} \psi_g(x, \mu), \quad (2a)$$

and its subsequent inversion

$$\psi_g(x, \mu) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dk e^{ikx} \bar{\psi}_g(k, \mu). \quad (2b)$$

In the transform k-space, Eq.(1) becomes

$$(\Sigma_g + ik\mu) \bar{\psi}_g(k, \mu) = \sum_{g'=1}^g \sum_{l=0}^L \frac{2l+1}{2} \Sigma_{l,g',g} P_l(\mu) \bar{\phi}_{g',l}(k) + Q(\mu) \delta_{g,1}, \quad (3)$$

where

$$\bar{\phi}_{g,l}(k) \equiv \int_{-1}^1 d\mu P_l(\mu) \psi_g(x, \mu).$$

Eq. (3) can be recast in the following form for the moments $\bar{\phi}_{g,l}$:

$$\sum_{l=0}^L \left\{ \delta_{j,l} - (2l+1) \Sigma_{l,g,g} T_{j,l}^g(k) \right\} \bar{\phi}_{g,l}(k) = \sum_{g'=1}^{g-1} \sum_{l=0}^L (2l+1) \Sigma_{l,g',g} T_{j,l}^g(k) \bar{\phi}_{g',l}(k) + \bar{Q}_j \delta_{g,1}, \quad (4)$$

with

$$T_{j,l}^g(k) \equiv \frac{1}{2} \int_{-1}^1 d\mu \frac{P_l(\mu) P_j(\mu)}{\Sigma_g + ik\mu},$$

$$\bar{Q}_j(k) \equiv \int_{-1}^1 d\mu \frac{P_j(\mu) Q(\mu)}{\Sigma_1 + ik\mu}.$$

The solution to Eqs. (3) in matrix form is

$$\bar{\phi}_g = A_g^{-1} \bar{q}_g, \quad (5)$$

where the elements of $\vec{\phi}_g$, $\vec{A}_g$ and $\vec{Q}_g$ are respectively,

$$\begin{aligned} \phi_{g,i} &= (2i+1) \Sigma_{\ell,g} T_{j,i}^g(k), \\ \text{and} \quad \sum_{g'=1}^{g-1} \sum_{\ell=0}^L (2\ell+1) \Sigma_{\ell,g',g} T_{j,i}^g(k) \bar{\phi}_{g',i}(k) &+ \bar{Q}_{g,i} \end{aligned}$$

The desired scalar flux is therefore

$$\phi_{g,0}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dk e^{ikx} \bar{\phi}_{g,0}(k). \quad (6)$$

Of course, one could attempt to evaluate Eq. (6) by complex contour methods, but for a cross section dependent on energy, this would be a hopeless task for all but a few group calculation. For this reason, a direct numerical implementation of the inversion is performed. Briefly, evaluation of Eq. (6) makes use of the equivalent form

$$\phi_{g,0}(x) = \frac{1}{\pi} \int_0^{\infty} dk \operatorname{Re}[\bar{\phi}_{g,0}(k)] \cos(kx). \quad (7)$$

The integral is then expressed as an infinite series with each term being an integration over either the half or full periods of the cosine. Each integral is numerically performed using an efficient Romberg integrator. Finally, the series is numerically summed via the Euler-Kuopp transformation to accelerate convergence.³

III. S_n Method

The S_n code used in this study was written to serve primarily as a general purpose slab-geometry research code for electron transport studies, and as a test-bed for multigroup algorithms. It incorporates the same linear discontinuous and diamond difference algorithms as are found in ONETRAN⁴. For simplicity, the S_n code was written to apply only to slab geometry. The main departure of our code from ONETRAN is that the angular components, rather than the Legendre moments, of the flux and source and angular scattering kernels are retained throughout the entire calculation. This feature can enable straightforward incorporation of angularly dependent total cross sections and source terms which may be used to represent the presence of, for example, an externally applied electric field⁵. This code is currently being used in two areas of electron transport research: 1) calculation of ionization doses resulting from the incidence of gamma radiation on multilayered material media as may be typified by microelectronic devices; and 2) studies of low energy electron-phonon scattering in solids.

IV. Results

The quantity taken as the basis for comparison of the S_n and benchmark calculations is the scalar flux. The approach taken in the comparisons between multigroup S_n and benchmark calculations begins with a series of one-group calculations for a plane isotropic source imbedded in an essentially infinite aluminum medium in which the scattering cross section was taken to be linearly anisotropic. For this one-group case we investigated the behavior of the ratio of successive errors as the spatial mesh size was decreased in order to reproduce the known theory. Then for the multigroup cases, S_n and benchmark calculations were made for a plane isotropic source imbedded in an infinite aluminum medium. These calculations, carried out using 40

uniformly spaced electron energy groups, were made for a 200 keV electron source in aluminum. Four scattering cases, corresponding to progressively higher scattering anisotropy order, were investigated. In the first, an artificial isotropic scattering cross section was devised, and in the remaining four, S_n and benchmark calculations were compared for extended transport corrected cross sections of order 2 through 4.

Because the benchmark calculations can only accommodate an infinite medium, and the S_n algorithm requires a finite medium, we chose the dimensions of the test mesh portions of the scattering media to be deeply imbedded within two very thick coarsely meshed slabs each of thickness 10 mean free paths. For the multigroup calculations, we chose the slab dimensions to correspond to ± 1 r.u. (= range unit), which for our 200 keV electron source in aluminum translates into a thickness of ± 0.058 g/cm². These geometries closely approximate infinite media for all practical purposes.

One-Group Calculation:

Plane Isotropic Source; Linearly Anisotropic Scattering

A set of 5 S_{12} calculations were made with the linear discontinuous differencing algorithm for a plane isotropic source located at the center ($x=0$) of a thick scattering medium extending 20 mean free paths in both positive and negative x -directions. The scattering parameters were (in units of mean free path), $\Sigma_t = 1.0$, $\Sigma_{s0} = 0.8$, $\Sigma_s = \Sigma_{s0}/3$. The spatial mesh discretization step sizes for the 5 calculations were taken to be $\Delta x_n = 2^{2-n}$; $n=1,2,3,4,5$. In each calculation, Δx was held constant over the spatial mesh test region ($-10 \leq x \leq +10$). Figure 1 is a plot of the successive error ratios $r_n(x)$ [$\equiv \epsilon_n(x)/\epsilon_{n+1}(x)$]. From the curves shown, a region of uniformity ($3 \leq |x| \leq 6$) could be identified for successive discretization error ratios. In this region it was found that the error always decreased for $n=1,2,3$. For the $n=1$ and $n=2$ cases, the ratio is nearly constant and has the value ~ 8 , which indicates third order accurate behavior for the linear discontinuous algorithm. Also in this region, when $n=3$, it is seen that the error ratio settles down to ~ 8 for one-half of the interval; however, the ratio r_4 does not show a consistent decrease in error as Δx is decreased from 0.25 to 0.125. This is attributable to the fact that it is not possible to further improve the result by halving Δx with a fixed inner iteration convergence precision. The fractional errors observed in these cases were generally comparable to the convergence precision (10^{-5}). Large variations in r_n were observed both near the source plane and the region boundary. The poor results near the source plane arise from the singular nature of the source geometry, while near the region boundary, changes in mesh size propagate errors into the test zone. The uniform region is situated well away from both.

Multigroup Calculations:

Plane Isotropic Source; Isotropic Scattering

Four sets of S_n calculations were made for the 200 keV isotropic electron source imbedded in infinite aluminum. The scattering cross section employed for these calculations is physically unrealistic in that 200 keV electrons in aluminum do not scatter isotropically. However in this first attempt to test the multigroup benchmark, we chose for convenience a cross section which contains only the zero-th order Legendre coefficient of the P-12 extended transport corrected Mott cross section (subsequent calculations, reported in the next section, make use of the actual P-2,3,4 extended transport corrected cross sections). These calculations covered the following four cases: linear discontinuous and diamond differencing, each used with S_6 and S_{12} Gauss-Legendre quadrature sets. In each of these four cases, 5 separate S_n calculations were made, corresponding to a graduated scale of spatial mesh resolutions [$\Delta x = .02, .01, .004, .002, .001$ r.u.] spanning a distance of 0.2 r.u. from the source plane at $x=0$. The reasons for the restriction to 0.2 r.u. are twofold: 1) to eliminate vacuum boundary

effects in the S_n calculations; 2) to operate within the high accuracy range of the analytical benchmark solution.

In the following we show only S_6 results, since in all cases studied the difference between S_6 and S_{12} was found to be negligible. Figure 2 is a plot of the scalar flux as calculated by the benchmark method (solid lines) and the S_n code (circles) with the diamond difference scheme for spatial resolution $\Delta x = .004$ r.u. The 10 curves are the energy spectra at $x = .02(.02).2$ r.u. As can be seen, the agreement is excellent.

Figure 3 is a plot of the exponent, b , of the relative error dependence [$\epsilon \propto (\Delta x)^{-b}$] of the spatial resolution vs. group midpoint energy, for 6 positions along x , as the resolution is increased from $\Delta x = .02$ to $\Delta x = .01$ r.u. for the linear discontinuous algorithm. The relative error (ϵ) is evaluated as the deviation of the S_n result from the benchmark result. The inner iteration convergence precision for the upper graph was set to 10^{-3} , while for the lower graph, we chose 10^{-5} . As anticipated, the exponent is -3 as dictated by theory. It should be noted that this holds true for a wider energy range as the accuracy of the S_n calculation is increased. However, at some point sufficiently low on the energy scale, no further improvement is achieved. As might be expected, the more accurate S_n calculation, to the extent that this crossover behavior exists at all, exhibits this behavior at a lower energy value. We believe that this is explained by an accumulation of the downscatter source error, which is not predominant in the high energy groups. It can also be noted that tightening of the convergence precision mitigates most of the erratic behavior of the exponent at the two positions ($x = .02, .06$ r.u.) closest to the source plane.

Plane Isotropic Source; Anisotropic Scattering

The 200 keV electron source (at $x=0$) and aluminum scatterer configuration ($|x| \leq 1.0$ r.u.) chosen for the anisotropic scattering benchmark tests were the same as those used in the previous case. It has already been demonstrated^{6,7} that a realistic description of the transport, or at least the resultant energy deposition, of 200 keV electrons in aluminum can be achieved using a P-12 extended transport corrected screened-Rutherford scattering cross section in conjunction with the S_n method. Doubtless, the use of lower order transport corrections for this same problem will result in a sacrifice of physical realism. However successive application of the extended transport correction from P-1 to P-4 solves the "same" problem if only very approximately, and additionally provides us with a means for systematically increasing the scattering anisotropy order of the benchmark calculations. Figures 4-6 show plots of the scalar flux at 10 equispaced x values $[0.04(0.04)0.4]$ for the P-2, P-3 and P-4 extended transport corrected cross sections (the L -th Legendre coefficient of the P-L extended transport cross section is zero, hence these represent scattering anisotropy orders 1,2,3). In all cases the linear discontinuous differencing algorithm was employed in 40 group- S_{12} calculations. The spatial discretization step was held constant at $\Delta x = 0.002$ r.u. As before, the solid curves represent the benchmark solution. The agreement between the two methods ranges between the third and fifth significant figure for all points except at the source plane.

V. Concluding Remarks

We believe that this investigation has served as a demonstration of the first multigroup benchmark transport solution. The excellent agreement between multigroup S_n and the benchmark results provides verification of the predicted error behavior for both the linear discontinuous and diamond difference schemes.

VI. References

1. J. E. Morel, *Nucl. Sci. Eng.*, **79**, 340 (1981).
2. S. Woolf, W. L. Filippone, B. D. Ganapol and J. C. Garth, *Nucl. Sci. Eng.*, **92**, 110 (1986).
3. B. D. Ganapol *et al.*, *Ann. Nucl. Ener.*, **17**, 49 (1990).
4. T. R. Hill, "ONETRAN: A Discrete Ordinates Finite Element Code for the Solution of the One-Dimensional Multigroup Transport Equation", LA-5990-MS, Los Alamos National Laboratory (1975).
5. B. R. Wienke, *J. Quant. Spect. Rad. Trans.*, **28**, 311 (1982).
6. J. E. Morel, *Nucl. Sci. Eng.*, **71**, 64 (1979).

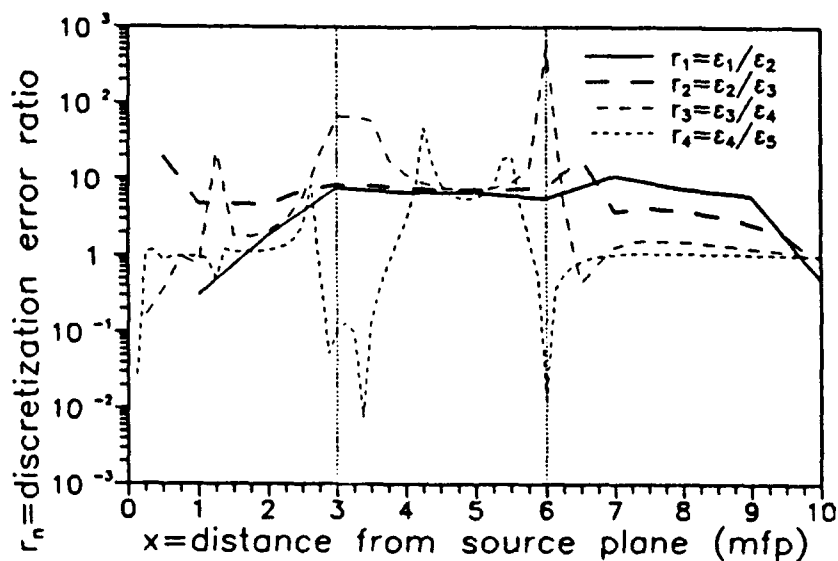


Fig. 1. Ratios of successive errors for 5 one-group linear discontinuous S_{12} calculations. Spatial discretization step size is varied according to $\Delta x_n = 2^{2-n}$; $n=1-5$. Stable region, $3 \leq |x| \leq 6$ (mean free path units), is indicated.

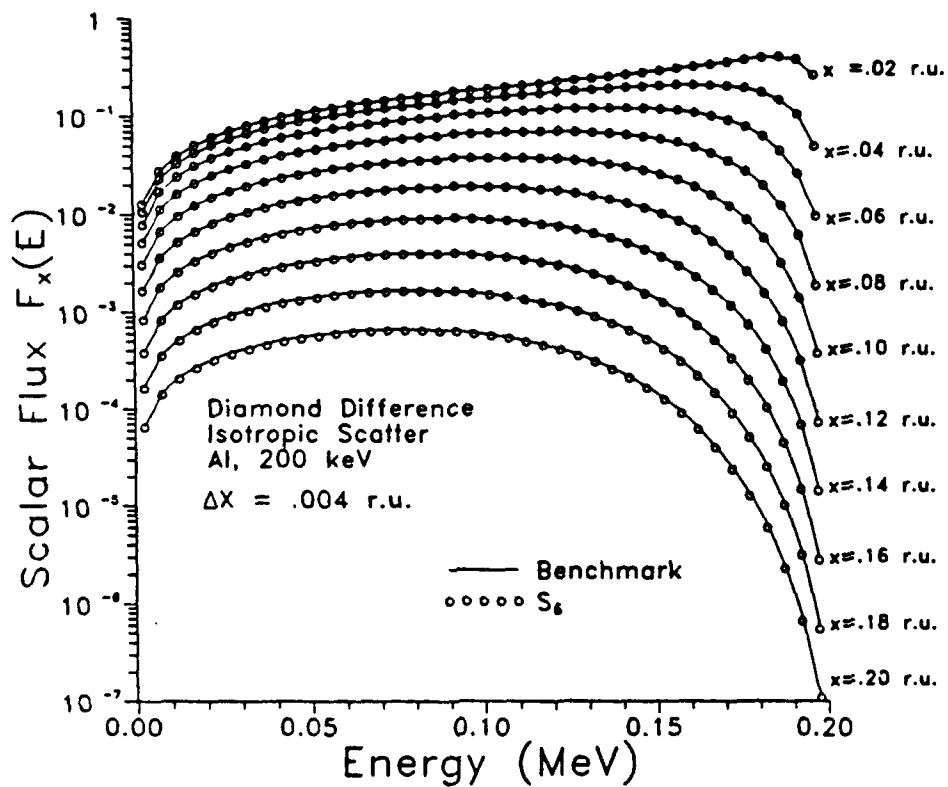


Fig. 2. Comparison of the scalar flux, $F_x(E)$ vs. E , for 40 energy groups at 10 x positions as indicated. Solid lines represent the benchmark calculation results; circles represent the results of a diamond difference S_6 calculation. Isotropic scattering.

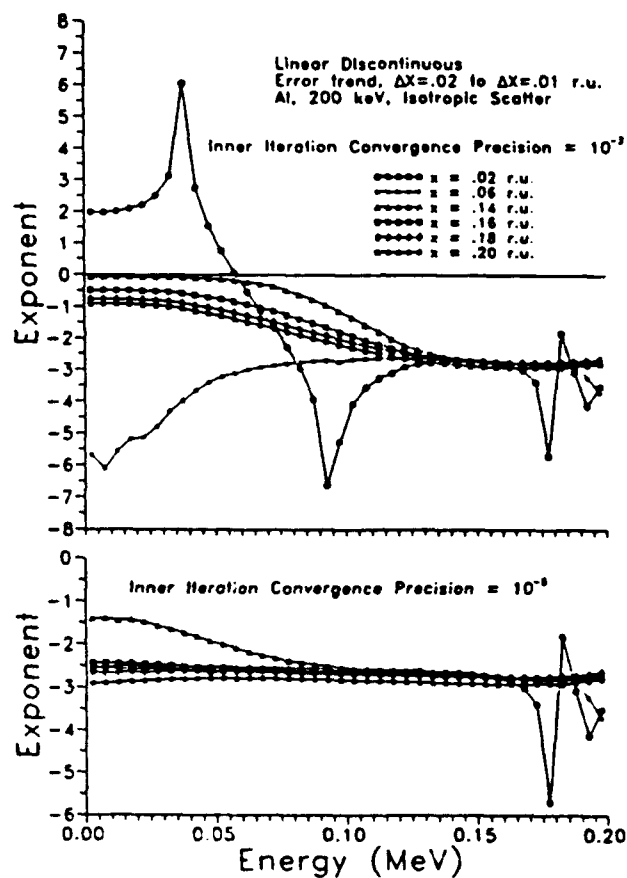


Fig. 3. Plots of the exponent $[b]$ of the relative error $[e]$, where $e \propto \Delta x^{-b}$, as Δx is decreased from .02 to .01 r.u. The behavior of b is shown as a function of energy at the six x positions indicated for two linear discontinuous S_8 calculations with inner iteration convergence precisions given by 10^{-3} (upper graph) and 10^{-5} (lower graph).

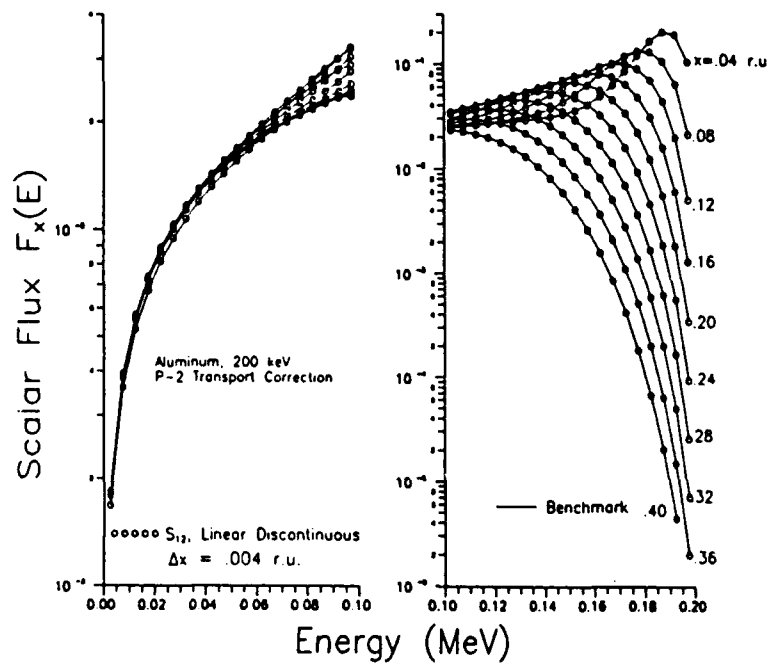


Fig. 4. Comparison of the scalar flux, $F_x(E)$ vs. E , for 40 energy groups at 10 x positions as indicated. Solid lines represent the benchmark calculation results; circles represent the results of a linear discontinuous S_{12} calculation. Anisotropic scattering of 200 keV electrons in Al with P-2 extended transport corrected cross sections.

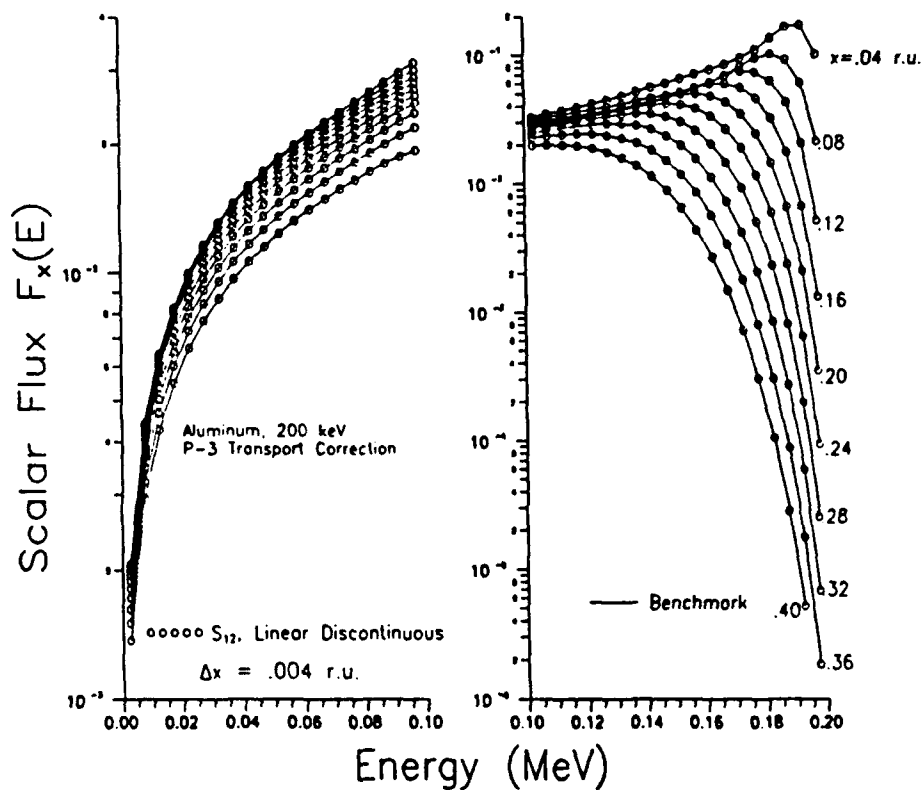


Fig. 5. Comparison of the scalar flux, $F_x(E)$ vs. E , for 40 energy groups at 10 x positions as indicated. Solid lines represent the benchmark calculation results; circles represent the results of a linear discontinuous S_{12} calculation. Anisotropic scattering of 200 keV electrons in Al with P-3 extended transport corrected cross sections.

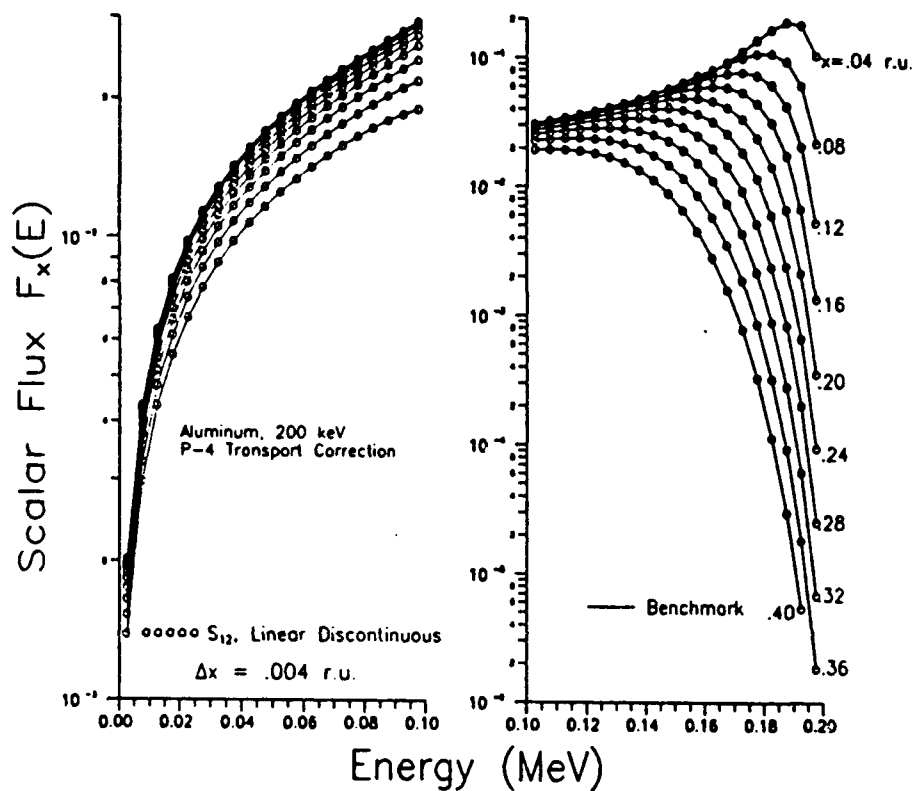


Fig. 6. Comparison of the scalar flux, $F_x(E)$ vs. E , for 40 energy groups at 10 x positions as indicated. Solid lines represent the benchmark calculation results; circles represent the results of a linear discontinuous S_{12} calculation. Anisotropic scattering of 200 keV electrons in Al with P-4 extended transport corrected cross sections.

38.1 9-11

U.S. GOVERNMENT PRINTING OFFICE

C11

**MISSION
OF
ROME LABORATORY**

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C³I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C³I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.