

AD-A048 614

RADIATION RESEARCH ASSOCIATES INC FORT WORTH TEX
STUDIES IN APPLICATION OF DISCRETE ORDINATES TRANSPORT METHODS --ETC(U)
SEP 77 D LINDSTROM

F/G 20/6

F08606-77-C-0008

UNCLASSIFIED

RRA-T7704

AFTAC-TR-78-1

NL

1 of 2

ADA048 614



AD A048614

AFTAC-TR-78-1

STUDIES IN APPLICATION OF DISCRETE
ORDINATES TRANSPORT METHODS TO
LIGHT TRANSPORT CALCULATIONS:
PREPARATION OF MIE THEORY CROSS
SECTIONS IN LEGENDRE POLYNOMIALS

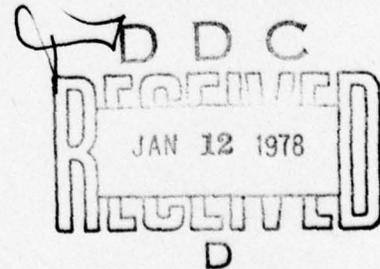


Radiation Research Associates, Inc.
Fort Worth, Texas

1 September 1977

Approved for public release, distribution unlimited

AIR FORCE TECHNICAL APPLICATIONS CENTER
HEADQUARTERS UNITED STATES AIR FORCE
PATRICK AIR FORCE BASE, FLORIDA 32925



AD INU.
DDC FILE COPY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER 18 AFTAC TR-78-1 ✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) 6 STUDIES IN APPLICATION OF DISCRETE ORDINATES TRANSPORT METHODS TO LIGHT TRANSPORT CALCULATIONS: PREPARATION OF MIE THEORY CROSS SECTIONS IN LEGENDRE POLYNOMIALS.		5. TYPE OF REPORT & PERIOD COVERED 9 Technical Report,	
7. AUTHOR 10 Duaine Lindstrom		8. PERFORMING ORG. REPORT NUMBER 14 RRA-T7704	
9. PERFORMING ORGANIZATION NAME AND ADDRESS RADIATION RESEARCH ASSOCIATES, INC. 3550 Hulen Street Fort Worth, Texas 76107 ✓		6. CONTRACT OR GRANT NUMBER(s) 15 F08606-77-C-0008 ✓	
11. CONTROLLING OFFICE NAME AND ADDRESS AIR FORCE TECHNICAL APPLICATIONS CENTER Patrick Air Force Base, Florida 32925		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 11 1 Sep 77	
		13. NUMBER OF PAGES 116 12 115p.	
		15. SECURITY CLASS. (of this report) Unclassified	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Mie Theory Discrete Ordinates Methods Light Scattering Legendre Polynomials			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A study was performed to determine the applicability of discrete ordinates codes to the transport of light through particulates. Light is scattered with high anisotropy by particulates. This high anisotropy requires expansion in high Legendre order for accurate expression by discrete ordinates codes. Procedures for convenient and economical production of light scattering data (Mie theory) in discrete ordinates format were developed and tested.			

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

294300

next page

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

One-dimensional time-dependent discrete ordinates calculations were performed using the TDA (Time Dependent ANISN) code with a simple cloud model. Computer storage and time limitations are outlined.

Extension to largest particle sizes and a second space dimension can best be accomplished using asymmetric discrete angle sets to conserve computer resources. Further economies might be accomplished by normalization of low-order data for problems dominated by multiple scattering. Relaxation of the need to express data in Legendre series could offer further economy, but may require significant development.

ACCESSION NO.	
DTIC	White Section <input checked="" type="checkbox"/>
DDC	Def. Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

DDC
RECEIVED
JAN 12 1978
D

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
LIST OF FIGURES	iv
LIST OF TABLES	v
I. INTRODUCTION	1
II. APPLICATION TO DISCRETE-ORDINATE TRANSPORT CODES	4
2.1 Geometry Limitations	4
2.1.1 One Dimension	4
2.1.2 Two Dimensions	6
III. THE USE OF LEGENDRE POLYNOMIALS IN THE BOLTZMANN TRANSPORT EQUATION	9
IV. CONVERSION OF MIE PARAMETERS TO LEGENDRE COEFFICIENT DATA	13
4.1 Legendre Coefficients for MIE Data	13
4.2 Legendre Coefficient Data Required	14
4.3 Application of Mie Theory Legendre Coefficients to ANISN	16
V. INFORMATION FLOW	18
VI. CROSS-SECTION ANALYSIS	22
VII. TEST EXPERIENCE - COMPUTER REQUIREMENTS	29
7.1 Cross-Section Generation	29
7.2 Angular Quadrature Order	29
7.3 Computer Storage Requirements	32
VIII. TEST EXPERIENCE - CALCULATIONAL RESULTS	37
REFERENCES	48
APPENDIX A. MIELEG UTILIZATION INSTRUCTIONS	49
APPENDIX B. LEGEDIT UTILIZATION INSTRUCTIONS	67
APPENDIX C. MACRO UTILIZATION INSTRUCTIONS	74
APPENDIX D. MACEDIT UTILIZATION INSTRUCTIONS	88
APPENDIX E. TDA UTILIZATION INSTRUCTIONS	92
APPENDIX F. FILE NAME DIRECTORY	108

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1.	Sample Case - Transmission from Point Source Through a Planar Cloud	5
2.	One-Dimensional Representation of Sample Case	7
	Two-Dimensional Representation of Sample Case	8
4.	Information Flow, Mie Theory Data Applied to Discrete Ordinates Transport	19
5.	Mie Theory Legendre Coefficients	23
6.	Effect of Coefficient Extraction by Refitting Weighted Function	24
7.	Relative Phase Function, Complete and Truncated Expansion	25
8.	Effect of Smoothing in Coefficient Refit	27
9.	Particle Density Function Assumed in Tests	30
10.	Computer Time to Produce Coefficient Set on LEG File	31
11.	Effect of S-Order on Error in P-35 TDA Calculation	33
12.	COMMON Array Storage Requirements for TDA Test Problem	35
13.	Running Time Requirements for TDA Test Problem	36
14.	Comparison of MIELEG- and MACRO-Generated Legendre Coefficients	39
15.	Effect of Coefficient Data Source on TDA-Calculated Fluxes	41
16.	Effect of Range of Particle Sizes on TDA-Calculated Fluxes	42
17.	Comparison of Cross-Section Normalizations, 0-4 Micron Particles	46
18.	Comparison of Cross-Section Normalizations, 0-6 Micron Particles	47

LIST OF TABLES

<u>Table</u>		<u>Page</u>
I.	PARAMETERS OF MACROSCOPIC DATA USED IN TEST CALCULATIONS	37
II.	TEST CASES FOR LOW-ORDER REPRESENTATION OF HIGH-ORDER SCATTER	43

I. INTRODUCTION

Discrete ordinates and other codes for radiative transfer calculations require phase function information to be expressed in terms of Legendre polynomials. A phase function, evaluated by Mie theory, can be expressed directly in Legendre polynomials by

$$p(\cos\theta) = \sum_{n=0}^N a_n P_n(\cos\theta) \quad (1)$$

where $P_n(\cos\theta)$ is the Legendre polynomial of order n and N^* is the maximum order necessary to sufficiently express the phase function and a_n is the associated Legendre coefficient.

The phase function $p(\cos\theta)$ is a function of scattering particle size, radiation wavelength, and refractive index. Thus, sets of a_n can be determined for each combination of size, wavelength, and refractive index. Normally, one would wish to perform transport calculations for a single wavelength and refractive index, but over a range of scattering particle sizes. An additional parameter is defined:

$$\alpha = 2\pi r/\lambda \quad (2)$$

where

$$\begin{aligned} \alpha &= \text{size parameter} \\ r &= \text{particle radius} \\ \lambda &= \text{radiation wavelength.} \end{aligned}$$

Then for a given refractive index and wavelength, a range of particle sizes is translated into a range of α -values. Sets of Legendre coefficients, a_n , can then be determined for each combination of refractive index and size parameter α .

* Mathematics tradition generally dictates the use of $n=0$ for the lowest order considered, but computer indexing usually does not allow the use of zero, so equations related to computer programming generally indicate $n=1, N$. Thus, the N of a computer-related equation may be 1 greater than N in an equivalent equation in traditional form.

There will generally be a different number of particles for each size, r . The number per cm^3 for any r is the particle size distribution $n(r)$ such that $\int n(r)dr$ is the density of particles, particles/ cm^3 , in the size range Δr . Appropriately averaged phase functions can be determined by integration of single size phase functions over an appropriate distribution $n(r)$.

Transport codes require data in the form of interaction probability per unit path length (macroscopic cross section) rather than the relative fractional scatter per particle given by the phase function. This interaction probability can be expressed in the same expansion form as the phase function:

$$\Sigma_s(\cos\theta) = \sum_{n=0}^N A_n P_n(\cos\theta) \quad (3)$$

where the coefficients, A_n are related to the phase function coefficients, a_n by

$$A_n = 10^{-8} \int_{\Delta r} \sigma_s(\alpha, \beta) a_n(\alpha, \beta) n(r) dr \quad (4)$$

and where

$\sigma_s(\alpha, \beta)$ = microscopic scattering cross section

α = size parameter, $2\pi r/\lambda$

$\beta = \alpha(m-ik)$

m = index of refraction relative to the surrounding medium

k = extinction coefficient of the particle material

$i = \sqrt{-1}$, imaginary coefficient for the complex number m

$n(r)$ = particle number density function for particles of size r

$10^{-8} = \text{cm}^2/\text{micron}^2$.

A set of procedures for calculating interaction probability coefficients has been produced and tested with the Time-Dependent ANISN (TDA) one-

space-dimension discrete ordinates code (Ref. 1). A description of the procedures and recommendations based on the tests follows. The appendices give utilization instructions for the computer codes used in this study and a file-name directory for the files stored on the Xerox Sigma-9 computer.

II. APPLICATION TO DISCRETE-ORDINATE TRANSPORT CODES

The use of Legendre expansions of cross section is widely practiced in computer calculations of radiation transport. Computer codes for large radiation transport calculations can generally be grouped into two categories: (1) Monte Carlo, based on random representation of natural processes, and (2) Boltzmann balances, based on the Boltzmann equation. Generally, the Monte Carlo method allows a wide choice of methods by which the cross-section data can be represented. The Boltzmann balance method codes, however, usually use cross sections represented in Legendre polynomials. Prominent among the Boltzmann balance codes are those using the S_n -discrete ordinate numerical techniques.

The S_n -discrete ordinate codes are limited to either one or two spatial dimensions, and the choice between them depends on the geometry of the case being modeled.

2.1 Geometry Limitations

2.1.1 One Dimension

One-dimensional codes (ANISN (Ref. 2), TDA, DRF (see Ref. 2)) are severely limited in the extent to which they can be used to model real situations (see Fig. 1).

Slab Geometry: If the scattering medium is considered as an infinite slab, with the allowed dimension the slab thickness, then the source must be an infinite plane source. Some code versions allow a choice of source directions, but all require spatial infinity.

Spherical Geometry: If the scattering medium is a sphere, with the allowed dimension the radial thickness of the increment, a point source is possible. A point source, however, must be located at the center of the sphere. Location at any radius greater than zero causes it to become a shell source.

Cylindrical Geometry: If the scattering medium is an infinitely long cylinder, with the allowed dimension the radial thickness of the medium, the source can be an infinitely long line source at the center or a cylindrical

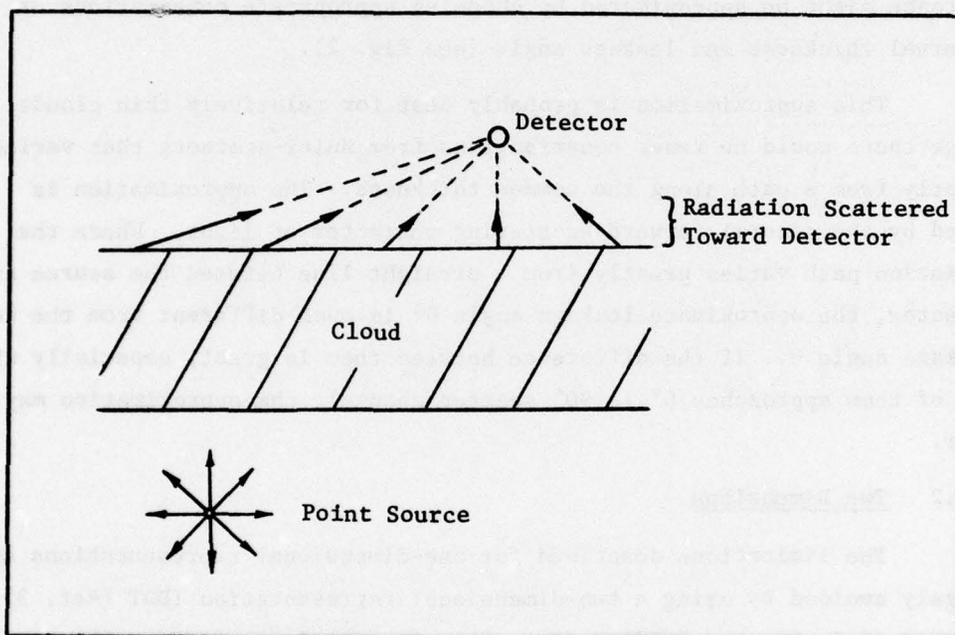


Fig. 1. Sample Case - Transmission from Point Source Through a Planar Cloud

shell source in any radial increment. The requirement of a point source necessarily limits one-dimensional simulation to spherical geometry. This, then, requires that the scattering medium have curvature equal to the distance from the source. This further requires that all of the source radiation be emitted normal to the incident cloud surface and that the detector position be determined by the radial distance from the source only. The real conditions of varying slant thickness and cloud-to-detector distance might be approximated by choosing appropriate combinations of interval thickness and leakage angle (see Fig. 2).

This approximation is probably best for relatively thin clouds, since there would be fewer contributions from multi-scatters that varied greatly from a path along the common thickness. The approximation is aided by the general forward-scattering character of light. Where the radiation path varies greatly from a straight line between the source and detector, the approximate leakage angle θ' is much different from the real leakage angle θ . If the difference between them is great, especially when one of them approaches 0° (a 90° scatter change), the approximation may be poor.

2.1.2 Two Dimensions

The limitations described for one-dimensional representations are largely avoided by using a two-dimensional representation (DOT (Ref. 3), TRANZIT (Ref. 4), and TWOTRAN (Ref. 5)). The sample case (Fig. 1), can be well-represented in cylindrical geometry (ρ, z) by describing a region of cloud around a normal from the source (see Fig. 3). The source itself can be represented by intervals in ρ and z around $\rho=0$. Then the detector response can be determined by integrating over all azimuthal positions in the upper surface r -intervals. Intervals can be defined to values of r as large as necessary to represent the detector response. The azimuthal symmetry assumed in the two-dimensional case is not a severe limitation as such symmetry can be reasonably assumed.

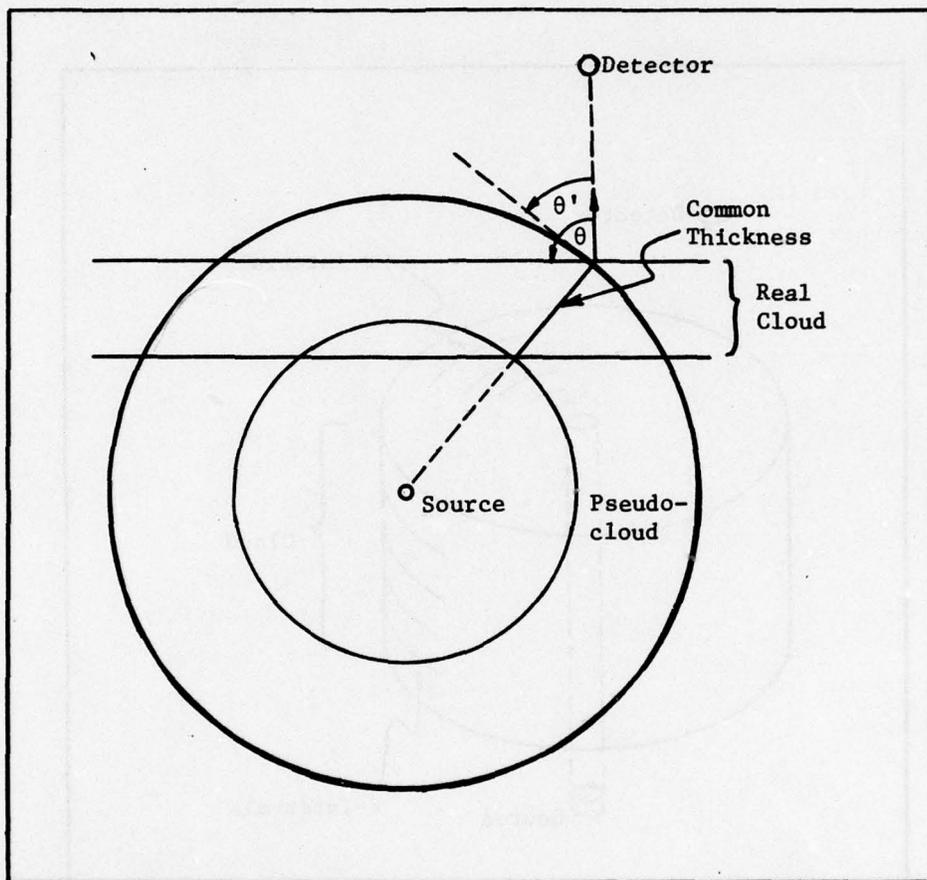


Fig. 2. One-dimensional Representation of Sample Case

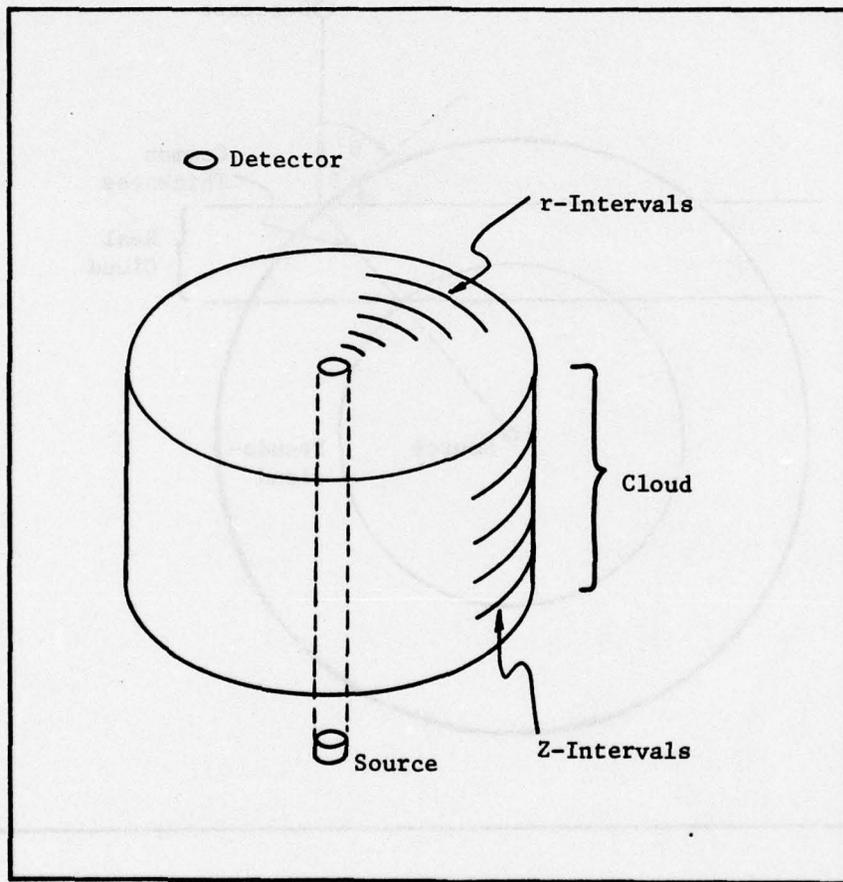


Fig. 3. Two-Dimensional Representation of Sample Case

III. THE USE OF LEGENDRE POLYNOMIALS IN THE BOLTZMANN TRANSPORT EQUATION

Many analytical and most numerical applications of the Boltzmann transport equation employ Legendre polynomials in some form. The reason for this lies in the form of the scattering term, which gives the density of particles (or photons) scattered into the direction $\bar{\Omega}$ from all other directions $\bar{\Omega}'$:

$$\int_{4\pi} \Sigma_s(\bar{\Omega}' \rightarrow \bar{\Omega}) \phi(\bar{\Omega}') d\Omega' \quad (5)$$

where $\phi(\bar{\Omega}')$ is the flux in direction $\bar{\Omega}'$. The scattering cross section, $\Sigma_s(\bar{\Omega}' \rightarrow \bar{\Omega})$, is then expanded in Legendre polynomials; for instance, by:

$$\Sigma_s(\bar{\Omega}' \rightarrow \bar{\Omega}) = \frac{1}{2\pi} \sum_{n=0}^N \left(\frac{2n+1}{2}\right) \Sigma_n P_n(\mu_0) \quad (6)$$

where Σ_n is the n +1st Legendre coefficient and $P_n(\mu_0)$ is the n +1st Legendre polynomial. The argument, μ_0 , is the cosine of the angle between the two vector directions, $\bar{\Omega}'$ and $\bar{\Omega}$. This expansion is very convenient for neutrons because, in many cases, isotropic scatter predominates. In isotropic scatter, the initial and final directions do not correlate and the probability of scatter into any unit solid angle is a constant. In this case, only the first term of the expansion needs to be used, where $P_n(\mu_0)$ is a constant ($P_0(\mu_0) = 1$). While this simplified case is seldom exact, it often performs well as a first approximation and good results are often obtained for relatively low orders of expansion (P_3 or P_5).

Scattering of gamma rays is not well-approximated by isotropic scattering, but is of sufficiently low order that good results are usually obtained by calculations employing only a few more terms than are used for neutrons. Light scattering, however, can be of a very high order when cross sections are expressed in Legendre polynomials. Orders of several hundred may be required to express Mie theory phase functions for relatively large aerosols (a few tens of microns in radius).

The convenience of the Legendre polynomial expansion in the scattering term lies in the form of the polynomial $P_n(\mu_0)$. It can be written in terms of the angles defining the initial and final directions, $\bar{\Omega}'$ and $\bar{\Omega}$.

$$\begin{aligned} P_n(\mu_0) &= P_n(\bar{\Omega}', \bar{\Omega}) \\ &= P_n(\mu)P_n(\mu') + 2 \sum_{\beta=1}^n \frac{(n-\beta)!}{(n+\beta)!} P_n^\beta(\mu) P_n^\beta(\mu') \cos\beta(\zeta-\zeta') \end{aligned} \quad (7)$$

where

$$P_n^\beta(\mu) = (-1)^\beta (1-\mu^2)^{\beta/2} \frac{d^\beta P_n(\mu)}{d\mu^\beta}$$

μ' and μ are the cosines of the polar angles describing the directions $\bar{\Omega}'$ and $\bar{\Omega}$,

ζ' and ζ are the azimuthal angles describing the directions $\bar{\Omega}'$ and $\bar{\Omega}$.

Then the scattering term can be written in two parts:

$$\frac{1}{2\pi} \sum_{n=0}^N \left(\frac{2n+1}{2}\right) \Sigma_n \int_0^{2\pi} d\zeta' \int_{-1}^1 d\mu' P_n(\mu) P_n(\mu') \phi(\bar{\Omega}') \quad (8)$$

and

$$\frac{1}{\pi} \sum_{n=0}^N \left(\frac{2n+1}{2}\right) \Sigma_n \int_0^{2\pi} d\zeta' \int_{-1}^1 d\mu' \sum_{\beta=1}^n \frac{(n-\beta)!}{(n+\beta)!} P_n^\beta(\mu) P_n^\beta(\mu') \cos\beta(\zeta-\zeta') \phi(\bar{\Omega}') \quad (9)$$

Assuming azimuthal symmetry in scattering, the first term becomes:

$$\sum_{n=0}^N \left(\frac{2n+1}{4\pi}\right) \Sigma_n P_n(\mu) \int_{-1}^1 d\mu' P_n(\mu') \phi(\mu') \quad (10)$$

The integral is simply a moment equivalent to a Legendre coefficient, ϕ_n in the flux expansion:

$$\phi(\bar{\Omega}') = \frac{1}{2\pi} \sum_{n=0}^N \left(\frac{2n+1}{2}\right) \phi_n P_n(\mu') \quad (11)$$

Therefore the first part becomes simply,

$$\sum_{n=0}^N \left(\frac{2n+1}{4\pi}\right) \Sigma_n P_n \phi_n \quad (12)$$

The second part can be rewritten to contain the integral

$$\int_0^{2\pi} d\zeta' \cos\beta(\zeta-\zeta'). \quad (13)$$

Making the substitution,

$$\alpha = \beta(\zeta - \zeta'), \quad (14)$$

the integral can be rewritten as

$$-\frac{1}{\beta} \int_{\beta(\zeta-2\pi)}^{\beta(\zeta)} \cos \alpha d\alpha = \frac{1}{\beta} \left[\sin \alpha \right]_{\beta(\zeta-2\pi)}^{\beta(\zeta)} = 0. \quad (15)$$

Therefore the entire scattering term reduces to the simple form of expression (12).

If it were desired to express the cross section, $\Sigma_s(\bar{\Omega}' \rightarrow \bar{\Omega})$ in some form other than Legendre polynomials, presumably one in which relatively low order were possible, but continued to express flux in Legendre polynomials, then the scattering term would take the form:

$$\begin{aligned} & \int_{4\pi} \Sigma_s(\bar{\Omega}' \rightarrow \bar{\Omega}) \left[\frac{1}{2\pi} \sum_{n=0}^N \left(\frac{2n+1}{2}\right) \phi_n P_n(\mu') \right] d\Omega' \\ & = \frac{1}{4\pi} \sum_{n=0}^N (2n+1) \phi_n \int_{4\pi} \Sigma_s(\bar{\Omega}' \rightarrow \bar{\Omega}) P_n(\mu') d\Omega' \end{aligned} \quad (16)$$

The moments defined by the integral would be different for each final direction considered. Thus, for instance, in a discrete-ordinate formulation using m quadratures, a total of m times n moments would be necessary. If this could be shown to be less than the present n necessary for Legendre expansion of cross section, the situation might be improved.

If a completely different orthogonal series were used to expand both cross section and flux, then an expression similar to (7) would have to be found for the series used or the expansion in cross section carried for all combinations of initial and final direction. That is, if the equivalent of $P_n(\mu_0)$ in the new polynomial were to be retained, then instead of using polynomials in terms of a single set of directions μ , polynomials would have to be evaluated in terms of all combinations of initial and final directions.

If m were the number of directions, m^2 instead of m polynomials would be used and n times m cross-section coefficients evaluated. As before, if this could be shown to be less than the n presently used in Legendre expansions, an improvement could result.

IV. CONVERSION OF MIE PARAMETERS TO LEGENDRE COEFFICIENT DATA

4.1 Legendre Coefficients for MIE Data

Legendre coefficients are calculated by MIELEG using the equation

$$a_n = \frac{2(-1)^n}{\alpha^2 K(\alpha, \beta)} \sum_{j=1}^{\infty} \sum_{k=1}^j \frac{2}{1+\delta_{jk}} \left[w_{jk} w_{jkn} + v_{jk} v_{jkn} \right] \quad (17)$$

from Ref. 6. The angular distribution function, $f(\theta)$ is related to these by

$$f(\theta) = \frac{1}{4\pi} \sum_{n=0}^{\infty} a_n P_n(x) \quad (18)$$

where $x = \cos\theta$ and $P_n(x) =$ nth Legendre polynomial, $f(\theta) =$ fraction scattered per unit solid angle about θ ; from which one should obtain

$$2\pi \int_0^{\pi} \sin\theta f(\theta) d\theta = \int_{-1}^1 f(x) dx = 1. \quad (19)$$

It is also related to the total power scattered of Mie theory:

$$i(\theta) = \frac{\lambda^2}{8\pi^2} (i_1 + i_2) \quad (20)$$

by the relation:

$$f(\theta) = \frac{i(\theta)}{\pi r^2 K} \quad (21)$$

$r =$ radius of the scattering particle

$K =$ scattering coefficient, σ_s / σ_g

$\sigma_s =$ scattering cross section

$\sigma_g =$ geometric cross section.

4.2 Legendre Coefficient Data Required

The differential scattering cross section, $\sigma(\theta)$ is the probability of scatter per unit solid angle about θ . Then the total probability of scattering (of a unit light flux per scattering particle) is

$$\begin{aligned}\sigma_s &= \int_0^\pi \sigma(\theta) d\Omega = 2\pi \int_0^\pi \sigma(\theta) \sin\theta d\theta \\ &= 2\pi \int_{-1}^1 \sigma(\theta) dx .\end{aligned}\tag{22}$$

It is desired that $\sigma(\theta)$ be expressed in terms of Legendre polynomials and a set of coefficients:

$$\sigma(\theta) = \sum_{n=0}^{\infty} C_n P_n(x)\tag{23}$$

or, to sufficient approximation,

$$\sigma(\theta) \approx \sum_{n=0}^N C_n P_n(x) .\tag{24}$$

These coefficients can be evaluated from $\sigma(\theta)$ information by integrating over appropriate polynomials:

$$\begin{aligned}\int_{-1}^1 \sigma(\theta) P_m(x) dx &= \sum_{n=0}^{\infty} C_n \int_{-1}^1 P_n(x) P_m(x) dx \\ &= C_m \left(\frac{2}{2m+1} \right)\end{aligned}\tag{25}$$

(by the orthogonality relationship of the Legendre polynomials.)

Thus
$$C_n = \left(\frac{2n+1}{2}\right) \int_{-1}^1 \sigma(\theta) P_n(x) dx. \quad (26)$$

Since the $n=0$ polynomial, $P_0(x)=1$, from Eq. (26),

$$C_0 = \frac{1}{2} \int_{-1}^1 \sigma(\theta) dx = \frac{1}{2} \left(\frac{\sigma_s}{2\pi}\right) = \frac{\sigma_s}{4\pi}. \quad (27)$$

A set of normalized coefficients σ_n can be defined by

$$\sigma_n \equiv 2\pi \left(\frac{2}{2n+1}\right) C_n \quad (28)$$

where $\sigma_0 = \sigma_s$.

Then
$$\sigma(\theta) \approx \frac{1}{2\pi} \sum_{n=0}^N \left(\frac{2n+1}{2}\right) \sigma_n P_n(x) \quad (29)$$

where
$$\sigma_n = 2\pi \int_{-1}^1 \sigma(\theta) P_n(x) dx. \quad (30)$$

Alternatively, the set could be defined by

$$\sigma'_n \equiv 4\pi C_n \quad (31)$$

in which case

$$\sigma(\theta) \approx \frac{1}{4\pi} \sum_{n=0}^N \sigma'_n P_n(x) \quad (32)$$

and
$$\sigma'_n = (2n+1)\sigma_n = 2\pi(2n+1) \int_{-1}^1 \sigma(\theta) P_n(x) dx. \quad (33)$$

This is the case assumed in ANISN and other Oak Ridge codes.

Thus, where the coefficients are defined by Eq.(28), they must be divided by $(2n+1)$ for input to codes assuming definition (31). Generally, Los Alamos codes (DTF, TRANZIT, TWOTRAN) assume definition (28).

4.3 Application of Mie Theory Legendre Coefficients to ANISN

In ANISN-related terms, the fraction of scattering in a solid angle about θ is

$$\frac{\sigma(\theta)}{\sigma_s} \approx \frac{1}{4\pi\sigma_s} \sum_{n=0}^N \sigma'_n P_n(x) \quad (34)$$

This is equivalent to $f(\theta)$ of Mie theory. From Eq.(18),

$$f(\theta) \approx \frac{1}{4\pi} \sum_{n=0}^N a_n P_n(x) \quad (35)$$

Therefore the coefficient required by ANISN, σ'_n , can be determined by equating the two relationships:

$$\frac{1}{4\pi} \sum_{n=0}^N a_n P_n(x) = \frac{1}{4\pi\sigma_s} \sum_{n=0}^N \sigma'_n P_n(x) \quad (36)$$

The equality is satisfied if, for all n ,

$$\sigma'_n = a_n \sigma_s \quad (37)$$

Since $K = \frac{\sigma_s}{\pi r^2}$, $\sigma_s = K\pi r^2$

so that $\sigma'_n = K\pi r^2 a_n$ *

* For use in LASL-type codes using the DTF format (defined by Eq. (30)), Mie theory coefficients must be divided by $(2n+1)$. This operation is carried out in the XSLA option of MACRO or MIELEG.

The Mie Legendre coefficients, a_n , are a function of particle size; or more correctly, of the size parameter, $\alpha = \frac{2\pi r}{\lambda}$, where λ is the radiation wavelength. Then for any distribution of particle sizes, $n(r)$, the coefficients used in the ANISN transport calculation must use appropriate averages over the distribution. The dimensions of σ'_n are those of microscopic cross section (distance)². Ultimately, ANISN requires data in dimensions of macroscopic cross section, (distance⁻¹). This is achieved by multiplying the microscopic cross section by scattering particle density. If the distribution, $n(r)$, is density per unit of particle size, such that

$$N = \int_0^{\infty} n(r) dr \quad (39)$$

gives the total particle density, then the appropriate macroscopic coefficients that ANISN can use directly in transport calculations are

$$\Sigma_n = \int_0^{\infty} \sigma'_n(\alpha) n(r) dr. \quad (40)$$

Or, for calculation of Σ_n where values of σ'_n are available for specific values of α ,

$$\Sigma_n \approx \sum_i \sigma_n(\alpha_i) n(r_i) \Delta r_i \quad (41)$$

where

Δr_i is the range of sizes represented by r_i and

$$\alpha_i = \frac{2\pi r_i}{\lambda} .$$

Expanded to angular cross section,

$$\Sigma_s(\theta) \approx \frac{1}{4\pi} \sum_{n=0}^N \Sigma_n P_n(\cos\theta). \quad (42)$$

V. INFORMATION FLOW

In general, the phase function coefficients, a_n , can be used to generate interaction probability coefficients, A_n , for different combinations of size and wavelength through the size parameter, α . Also, for large values of α , the computer time necessary to calculate values of a_n is sufficiently great that unnecessary recalculations should be avoided. Therefore, the procedures for preparing input data for a discrete ordinates code is built around a file of collected values of a_n . This file, LEG, can contain all values previously calculated and can be drawn upon to produce values of A_n for discrete ordinates input data. A diagram of the information flow between codes and files is shown in Fig. 4.

The MIELEG code is based on the MIE-2 code (Ref. 7) which calculates Mie Theory cross sections and phase functions, and the Legendre expansion expressions outlined by Clark, Chu, and Churchill (Ref. 8). This code calculates from r , λ , and β a set of values of a_n and adds them to the LEG file. Any other set of r, λ values leading to the same size parameter, α , would produce the same set. For a range of values of r , a number of sets can be produced, distributed across the range. In addition to saving these sets on the LEG file, if the parameters of a distribution function are input, MIELEG will produce a set of macroscopic parameters, A_n . These will be calculated from the approximation to Eq. (4):

$$A_n = 10^{-8} \sum_{i=1}^{NRS} K(\alpha_i, \beta) \pi r_i^2 a_n(\alpha_i, \beta) n(r_i) \Delta r_i \quad (43)$$

where

- i is the index of a single r ,
- NRS is the number of r values used,
- Δr_i is the interval of r associated with value r_i ,
- $n(r_i)$ is the relative density of particles of size r_i ,

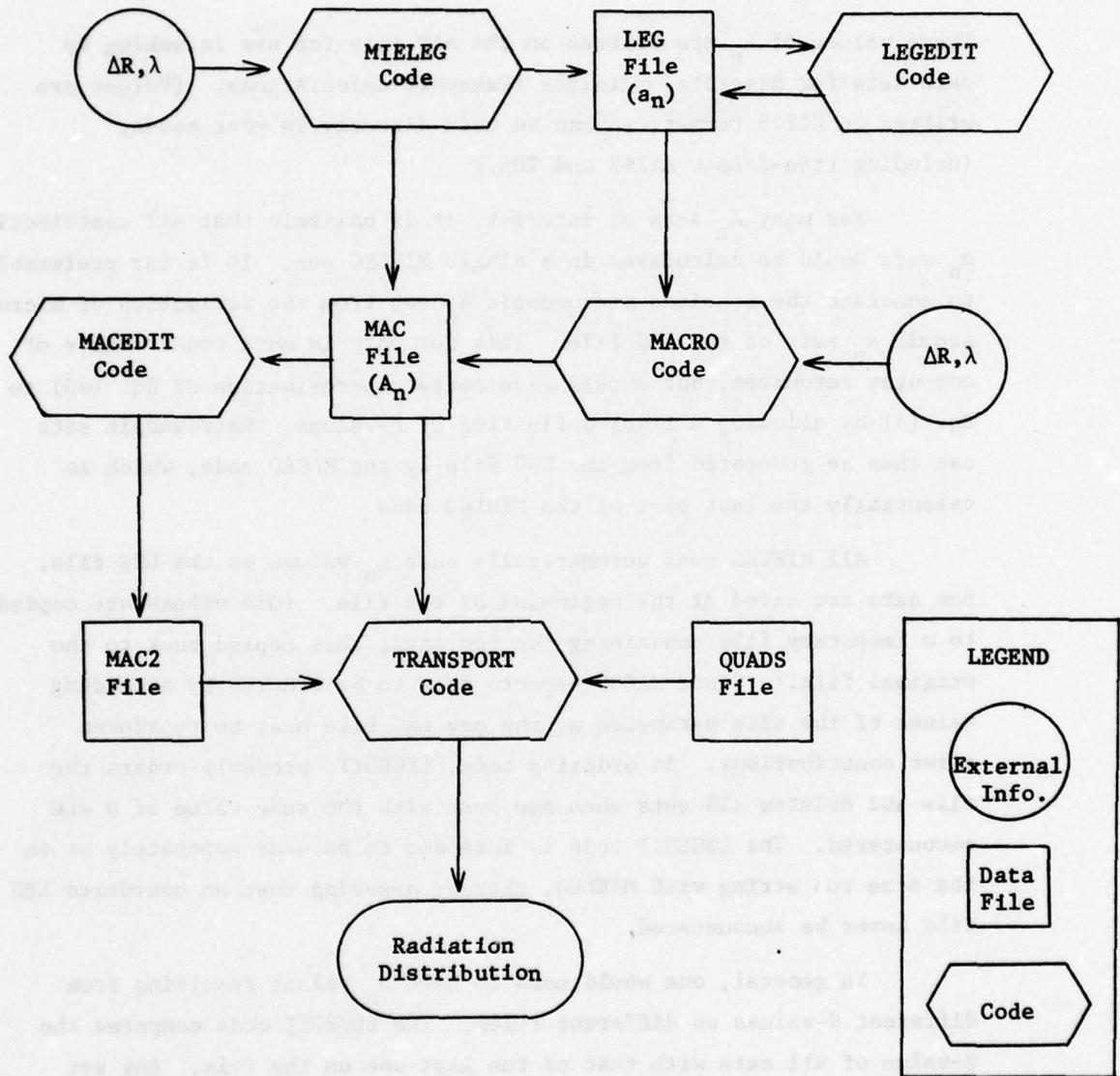


Fig. 4. Information Flow, Mie Theory Data Applied to Discrete Ordinates Transport

K is the cross-section ratio, $\sigma_s/\pi r_1^2$, and
 σ_s is the scattering cross section.

These values of A_n are written on the MAC file for use in making up data sets for discrete ordinates transport calculations. (Values are written on E12.5 format, so can be used directly in most codes, including free-format ANISN and TDA.)

For many A_n sets of interest, it is unlikely that all contributing a_n sets would be calculated in a single MIELEG run. It is far preferable to generate the required macroscopic A_n set from the collection of microscopic a_n sets on the LEG file. This not only is more conservative of computer resources, but should give better approximation of Eq. (43) to Eq. (4) by allowing a finer definition of r -values. Macroscopic sets can then be generated from the LEG file by the MACRO code, which is essentially the last part of the MIELEG code.

All MIELEG runs automatically save a_n values on the LEG file. New sets are saved at the beginning of the file. (Old values are copied to a temporary file containing the new ones, then copied back to the original file.) Since MACRO expects sets to be ordered by ascending values of the size parameter α , the new LEG file must be reordered after contributions. An ordering code, LEGEDIT, properly orders the file and deletes old sets when new ones with the same value of α are encountered. The LEGEDIT code is intended to be used separately or in the same run string with MIELEG, thereby assuring that an unordered LEG file never be encountered,

In general, one would need to have a_n values resulting from different β -values on different files. The LEGEDIT code compares the β -value of all sets with that of the last one on the file. Any set whose value does not agree with that of the last set is deleted and written on an auxiliary file, OTHERLEG. It is thus possible, by changing file assignments in the LEGEDIT job control sequence, to unfold a file containing sets with several β -values into several sets, each with a single β -value.

In ordinary production of a set of A_n values for use in a transport code, one would choose the LEG file containing the β -value of interest and input to the MACRO code the wavelength and size distribution required. MACRO will then produce a file, MAC, containing the set in E12.5 format. This file can be edited by CRT terminal to MERGE the A_n values into the transport code data file.

Mie theory Legendre coefficient sets are quite lengthy, requiring a much higher order than is necessary for transport of other radiations. Proper use of discrete ordinates codes generally requires S-orders (angular quadrature) greater than or equal to P-orders (Legendre). (Excessively large S-order may require very long computing times.) Thus, for convenience in changing quadrature sets (weights and cosines), a file, QUADS, has been established, into which all input sets can be read for merging into transport code input files.

A short normalizing routine, MACEDIT, has been included for data normalization in terms of the first-order coefficient A_0 . This is equivalent to the total scattering cross section and can be used to normalize the entire set. The normalized set (along with the original set from MAC) is saved on the MAC2 file.

All procedures were implemented on the SIGMA-9 computer at Texas Christian University, accessed by CRT terminal. All discussion is directed toward usage on this system.

VI. CROSS-SECTION ANALYSIS

A significant characteristic of Legendre expansions of Mie theory cross sections is the large order (i.e., number of coefficients) that is necessary for adequate representation. The order needed is about 2α , where α is the size parameter, $2\pi R/\lambda$. (Some authors suggest $2\alpha+2$.) If the expansion is performed to produce angular-dependent from order-dependent data (Eq. 1 or Eq. 3), a final term, N , greater than $2\alpha+2$ should not contribute significantly to the result. As examples of this characteristic, three sets of microscopic coefficients, a_n , produced by MIELEG are shown in Fig. 5.

An indication of the sufficiency of the expansion can be obtained by refitting the expanded function by integrating, for each order, the function multiplied by the Legendre polynomial of that order (in the manner of Eq. 24). In Fig. 6 is a plot of a weighted average of coefficients from $\alpha=8.46$ to $\alpha = 25.22$ (normalized to 1.0) and the coefficients obtained from a refit of the expanded function.

The refitted coefficients become effectively constant when the original coefficients become small. The calculation was performed using two different recursion relations for generation of the Legendre polynomials, and with both single and double precision in the polynomial. Since there were no significant differences in the resulting coefficients, one must assume that the nearly-constant values result from the availability of detail in the expanded function, rather than from round-off error.

Re-expansion from the refitted coefficients would result in little contribution from the constant coefficients since the sum of several polynomials is approximately zero. If the coefficients were to be obtained by fitting, as from MIE-2 phase functions, rather than directly from Mie Theory (Eq. 17), this "tail" of constant values would result. Judicious truncation of a set of coefficients obtained by either technique results in adequate representation of the function.

Early truncation of the series may give a greatly inadequate representation of the function. Figure 7 is a comparison of a full

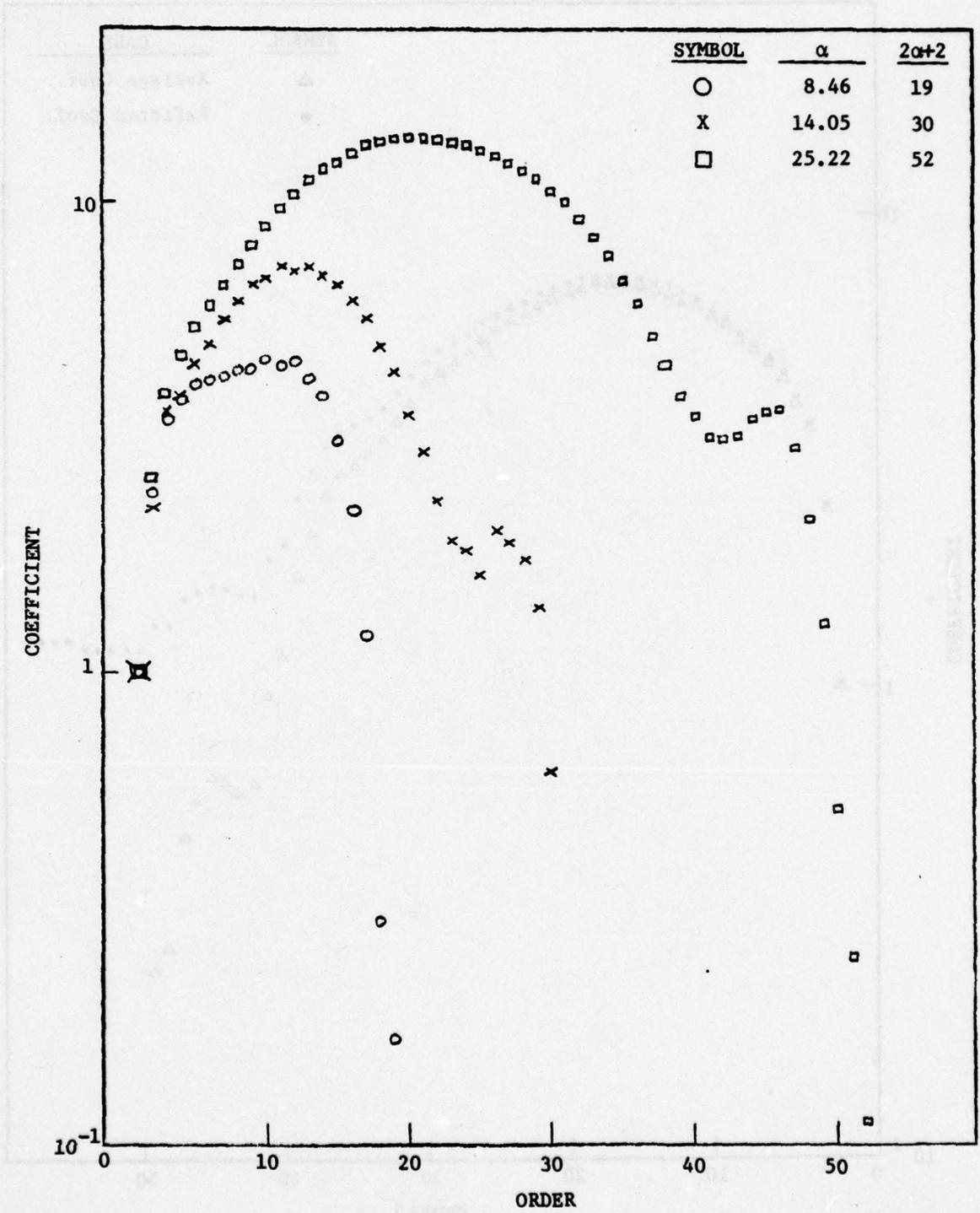


Fig. 5. Mie Theory Legendre Coefficients

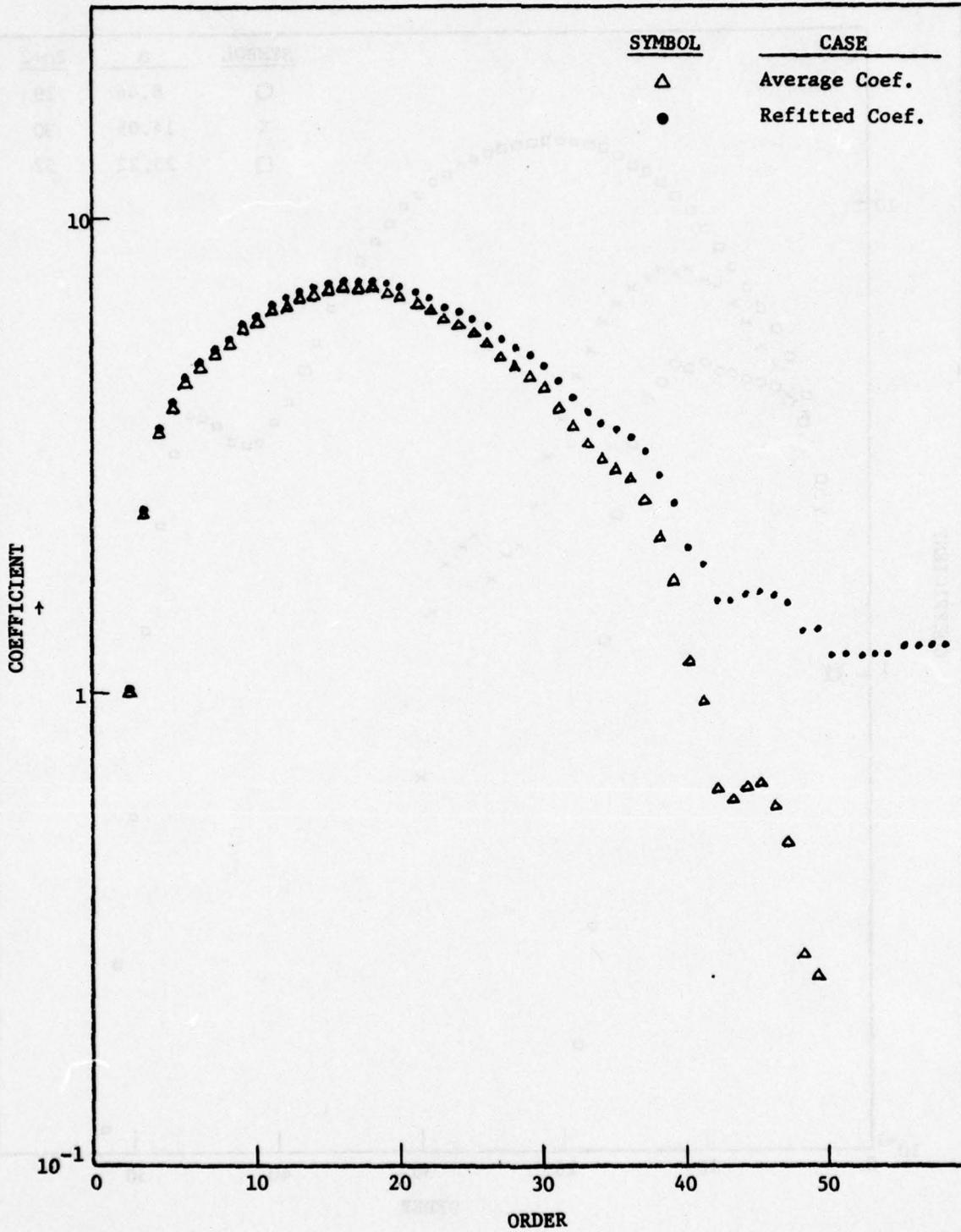


Fig. 6. Effect of Coefficient Extraction by Refitting Weighted Function

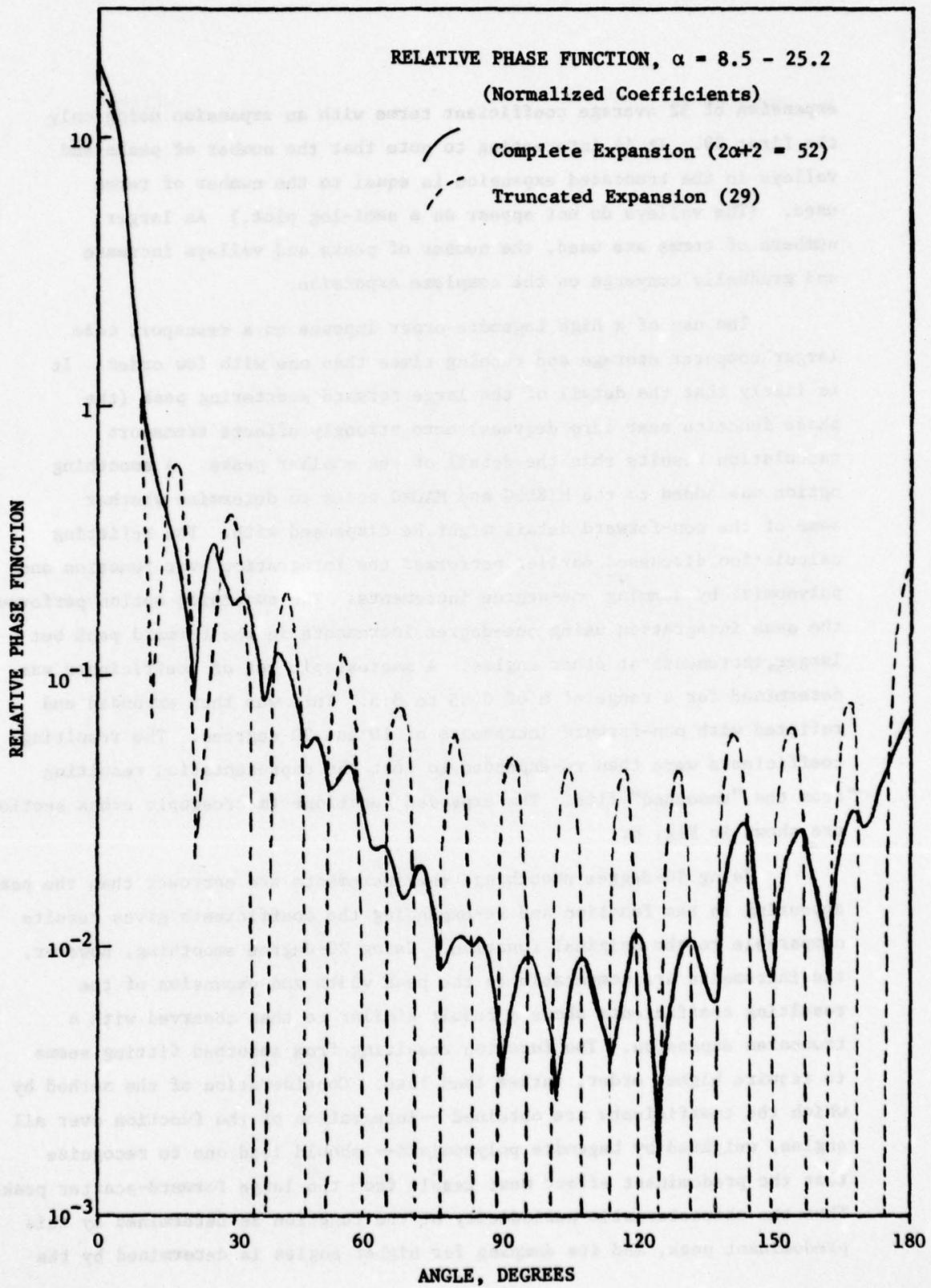


Fig. 7. Relative Phase Function, Complete and Truncated Expansion

expansion of 52 average coefficient terms with an expansion using only the first 29. It is interesting to note that the number of peaks and valleys in the truncated expansion is equal to the number of terms used. (The valleys do not appear on a semi-log plot.) As larger numbers of terms are used, the number of peaks and valleys increase and gradually converge on the complete expansion.

The use of a high Legendre order imposes on a transport code larger computer storage and running times than one with low order. It is likely that the detail of the large forward scattering peak (the phase function near zero degrees) more strongly affects transport calculation results than the detail of the smaller peaks. A smoothing option was added to the MIELEG and MACRO codes to determine whether some of the non-forward detail might be dispensed with. The refitting calculation discussed earlier performed the integration over function and polynomial by summing one-degree increments. The smoothing option performed the same integration using one-degree increments in the forward peak but larger increments at other angles. A macroscopic set of coefficients was determined for a range of α of 0.85 to 8.5. This was then expanded and refitted with non-forward increments of 10 and 20 degrees. The resulting coefficients were then re-expanded to test the representation resulting from the "smoothed" fits. The expanded functions (macroscopic cross sections) are shown in Fig. 8.

Using 10-degree smoothing, the increments are narrower than the peaks appearing in the function and re-expanding the coefficients gives results comparable to the original function. Using 20-degree smoothing, however, the increments are comparable to the peak width and expansion of the resulting coefficients shows a result similar to that observed with a truncated expansion. The function resulting from smoothed fitting seems to require higher order, rather than less. Consideration of the method by which the coefficients are obtained—integration of the function over all angles, weighted by Legendre polynomials—should lead one to recognize that the predominant effect must result from the large forward-scatter peak. Thus the characteristic periodicity of the function is determined by this predominant peak, and its damping for higher angles is determined by the

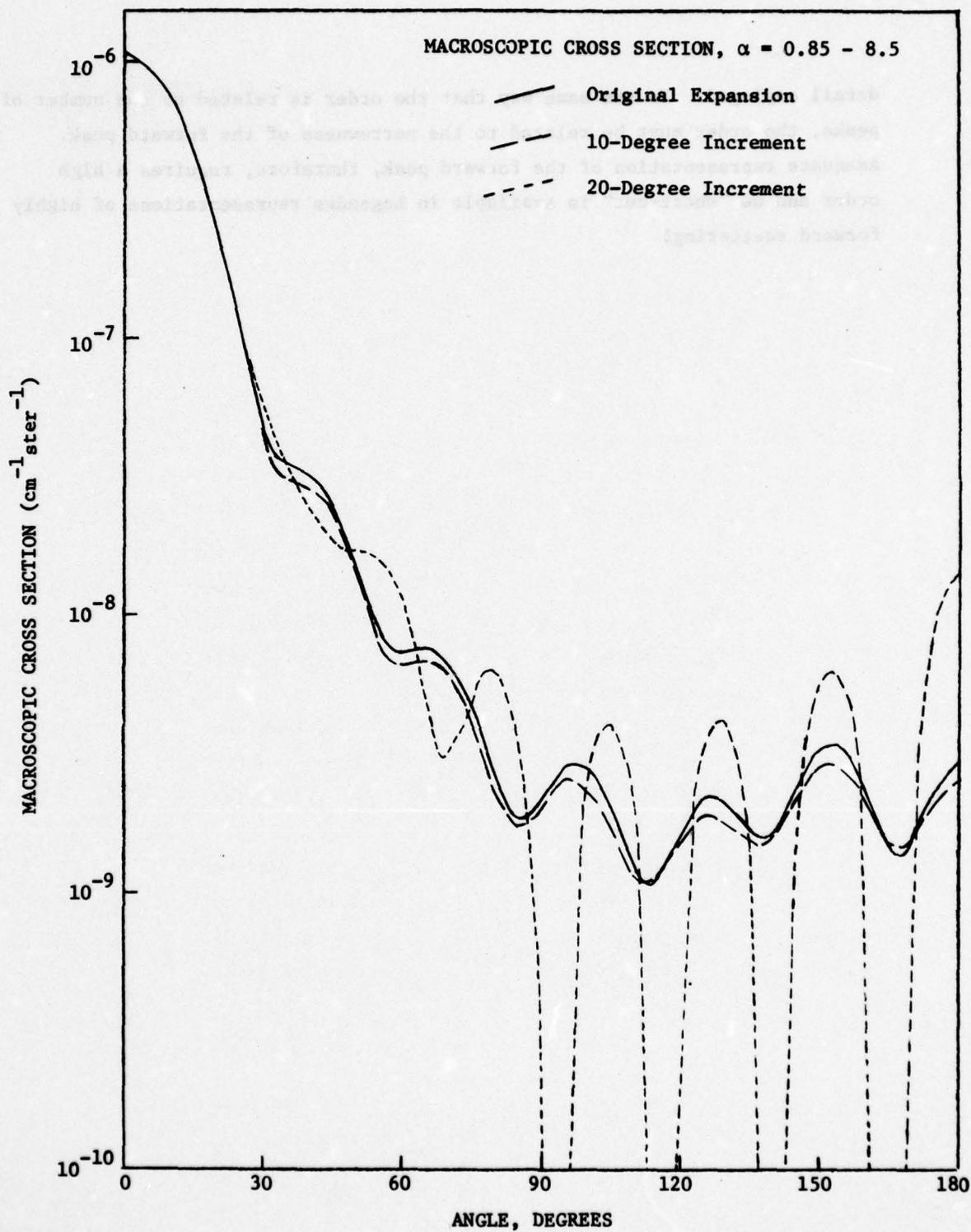
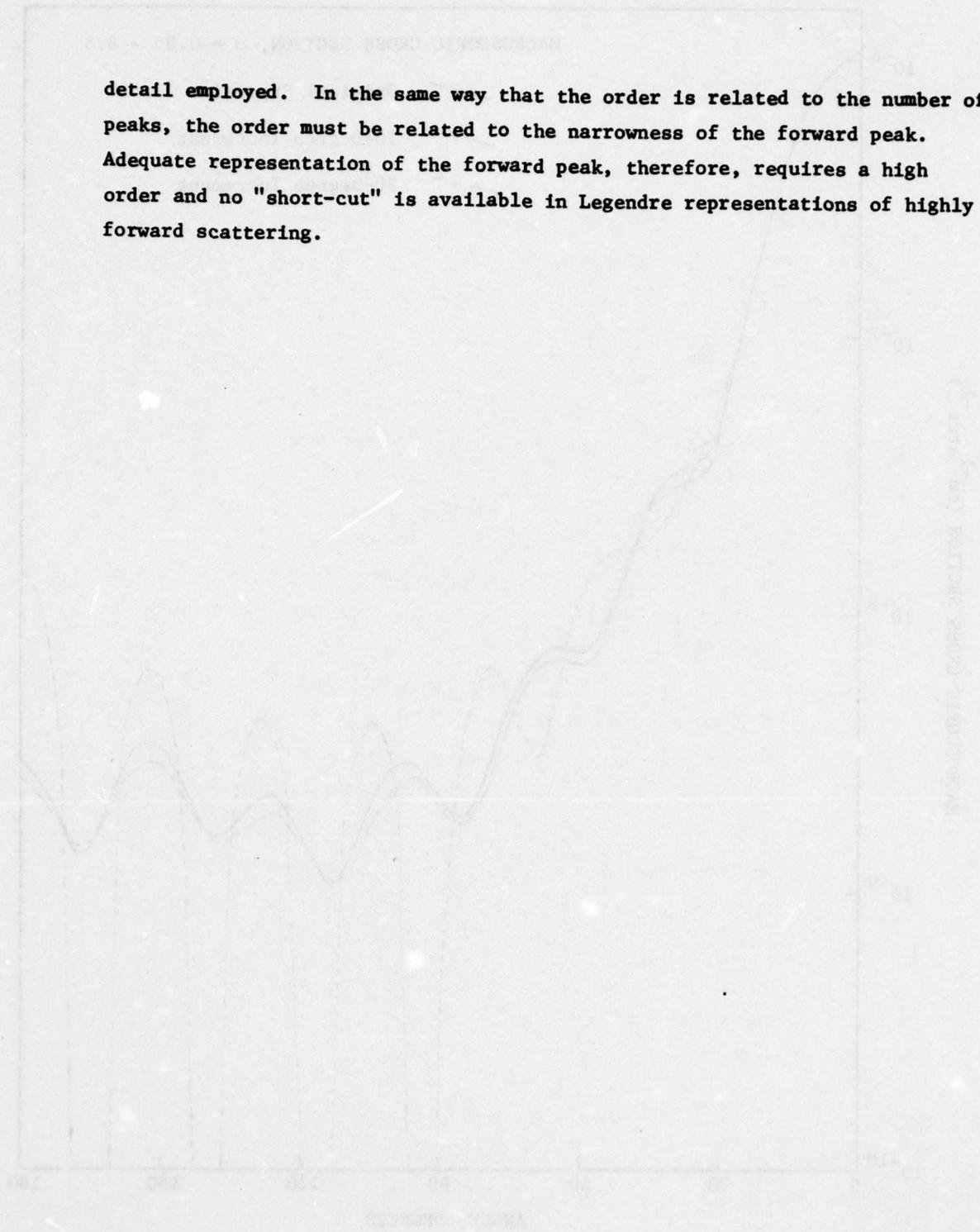


Fig. 8. Effect of Smoothing in Coefficient Refit

detail employed. In the same way that the order is related to the number of peaks, the order must be related to the narrowness of the forward peak. Adequate representation of the forward peak, therefore, requires a high order and no "short-cut" is available in Legendre representations of highly forward scattering.



VII. TEST EXPERIENCE - COMPUTER REQUIREMENTS

7.1 Cross-Section Generation

A series of cross-section sets was produced and tested on the Time Dependent ANISN (TDA) code. A case was modeled with a unit point source at the center of a 0.5-km radius cloud. Time increments summing to more than 10 μ sec were specified. (The unscattered wave front would reach the edge of the cloud in 1.67 μ sec). Early attempts to generate complete cross-section sets, using particle sizes as would be expected in a cloud, required long MIELEG running times (considerably more than the few minutes estimated). In addition, the large orders involved appeared to require large computer storage and CPU time with TDA. At this time it was determined to produce the LEG file and LEGEDIT and MACRO codes to conserve cross-section preparation time and to perform test calculations on ranges of particle sizes.

The particle density function used in the tests was given by the expression

$$n(R) = 27R^2 e^{-0.6R} \quad (44)$$

where R is given in microns (10^{-4} cm) and n(R) has units of particles cm^{-4} . The density function is shown in Fig. 9.

Microscopic coefficients were produced by MIELEG for 47 size parameters up to a value of 84, and saved on the LEG file. All assumed a complex refractive index of (1.33, 0.0). For a light wavelength of 0.75 microns, the LEG file can provide data for particle sizes to 10 microns. The CPU time as a function of size parameter, α , is shown on Fig. 10. All times include a following LEGEDIT run to order the file.

7.2 Angular Quadrature Order

The cross-section order (P-order) necessary to reliably represent Mie phase functions was determined earlier. A related parameter is the angular quadrature order (S-order) necessary to adequately represent the anisotropy of scattering in the balance equations of the discrete ordinates

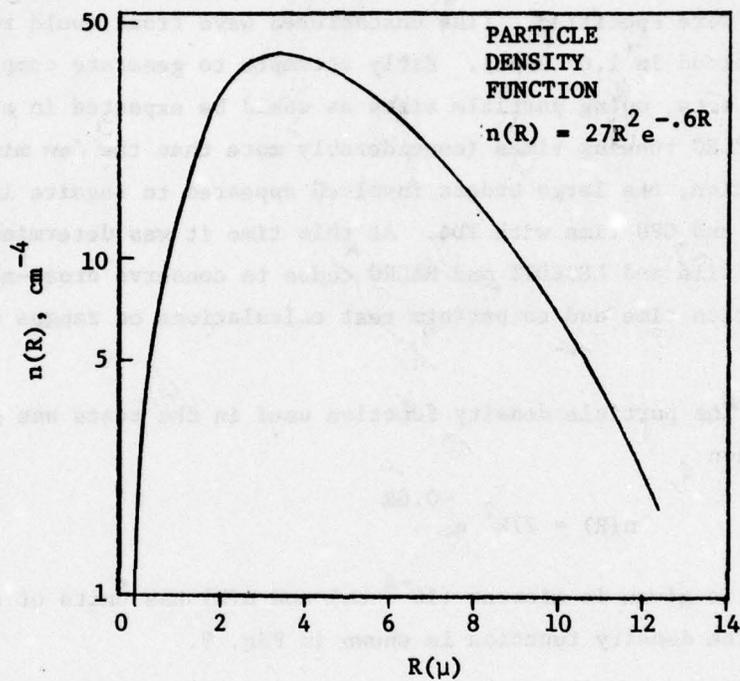


Fig. 9. Particle Density Function Assumed in Tests

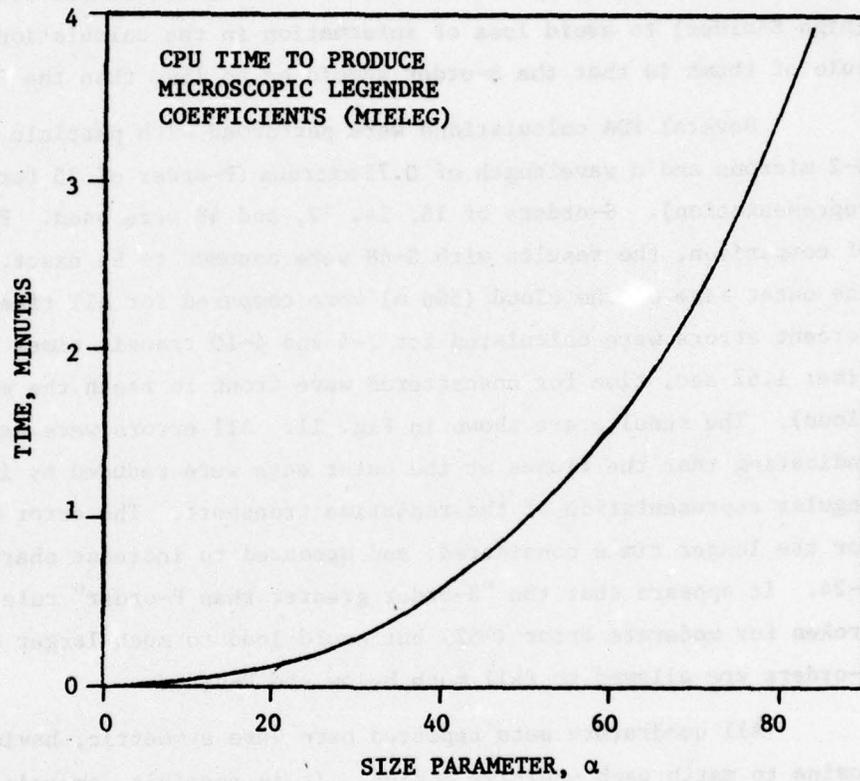


Fig. 10. Computer Time to Produce Coefficient Set on LEG File

calculations. The "quadratures" are the vector direction cosines, x , along which angular fluxes are calculated. Associated with the cosines are weights, w , representing the fractional solid angles around each vector direction. Taken together, these represent discrete directions (thus "discrete ordinates") that the radiation can take. Highly peaked cross sections generally require the use of a large number of directions (high S-order) to avoid loss of information in the calculation. A general rule of thumb is that the S-order should be no less than the P-order.

Several TDA calculations were performed with particle sizes of 0-2 microns and a wavelength of 0.75 microns (P-order of 35 for $2\alpha+2$ representation). S-orders of 16, 24, 32, and 48 were used. For the sake of comparison, the results with S-48 were assumed to be exact. Fluxes at the outer edge of the cloud (500 m) were compared for all time steps. Average percent errors were calculated for 1-4 and 4-10 transit times (1 transit time: 1.67 sec, time for unscattered wave front to reach the edge of the cloud). The results are shown in Fig. 11. All errors were negative, indicating that the fluxes at the outer edge were reduced by inadequate angular representation of the radiation transport. The error was greater for the longer times considered, and appeared to increase sharply below S-24. It appears that the "S-order greater than P-order" rule might be broken for moderate error ($\sim 5\%$) but could lead to much larger error if S-orders are allowed to fall much below the P-order.

All quadrature sets reported here were symmetric, having a negative cosine to match each positive cosine. It is possible, exercising great care, to use "asymmetric quadrature sets", which have a greater density of directions considered near $x=1.0$ to better represent the forward scattering. A single TDA run was made using such a quadrature set (of unknown origin). These results are not reported except to say that the fluxes were 20-50% greater than the S-48 fluxes, so should not be accepted as reliable without further study.

7.3 Computer Storage Requirements

Several other TDA jobs were run, with maximum particle sizes up to 6 microns, with P-orders up to 95 and S-orders from 16 to 48. Core storage

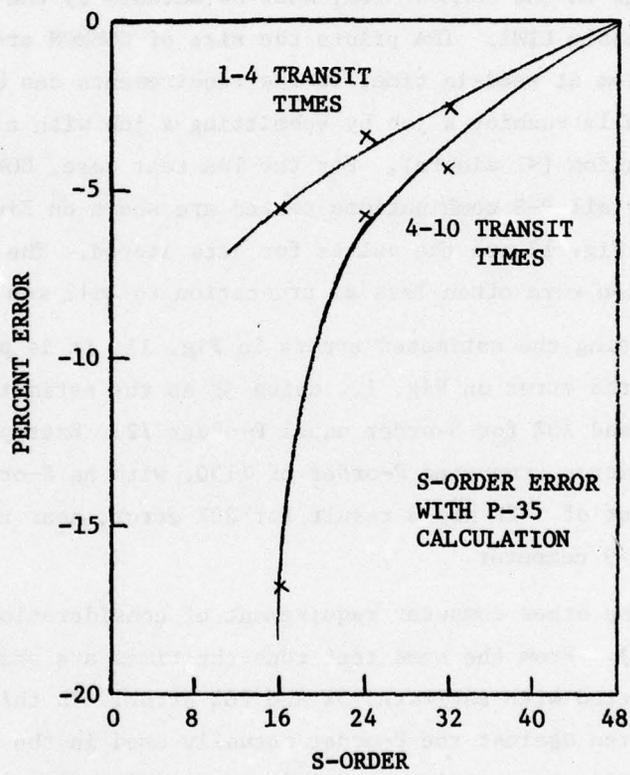


Fig. 11. Effect of S-Order on Error in P-35 TDA Calculation

requirements are directly related to the S- and P-orders. The MAIN routine of TDA contains a COMMON statement specifying the size of the flexibly-dimensioned storage array. Load modules containing array sizes of 20,000 (file name: TDA) and 30,000 (TDA30K) were used. It is likely that the largest array size that can be used with the TDA code and the SIGMA 9 computer is about 46,000 with a total core usage of 64,000. Any change in the COMMON array must be matched by the value of the size test variable LIM1. TDA prints the size of COMMON array necessary for any problem at read-in time, so the requirements can be determined without actually running a job by submitting a job with a very short run time specification (<1 minute). For the TDA test case, COMMON array requirements for all P-S combinations tested are shown on Fig. 12. (The P-values shown on Fig. 12 are the values for data stored. The P-values of the calculation were often less as truncation to $2\alpha+2$ was employed.)

Using the estimated errors in Fig. 11, it is possible to supply lines of estimated error on Fig. 12, using 5% as the estimate for S-order equal to P-order, and 20% for S-order equal P-order /2. Extrapolating these results to an 8-micron truncated P-order of >130, with an S-order of 64, a storage requirement of ~42K would result for 20% error, near the maximum of 46K for the Sigma-9 computer.

The other computer requirement of consideration is the running time (CPU time). From the same test runs the times are shown on Fig. 13, again cross-plotted with estimated 5% and 20% error. In this case, the values were plotted against the P-order actually used in the calculation. Extrapolation to a maximum particle size of 8 microns is more difficult for CPU time than it was for COMMON storage, but it appears that a run time of more than 100 minutes for 20% error is likely.

Successful use of asymmetric quadrature sets could reduce the effective S-order by a factor of 3 or more while retaining the degree of error. This could reduce the storage requirements to $\sim 1/2$ and time to $\sim 1/3$ for the test cases reported here. It is likely, then, that use of asymmetric quadrature sets could extend the range of solution to 12 microns or more, effectively covering the particle size range.

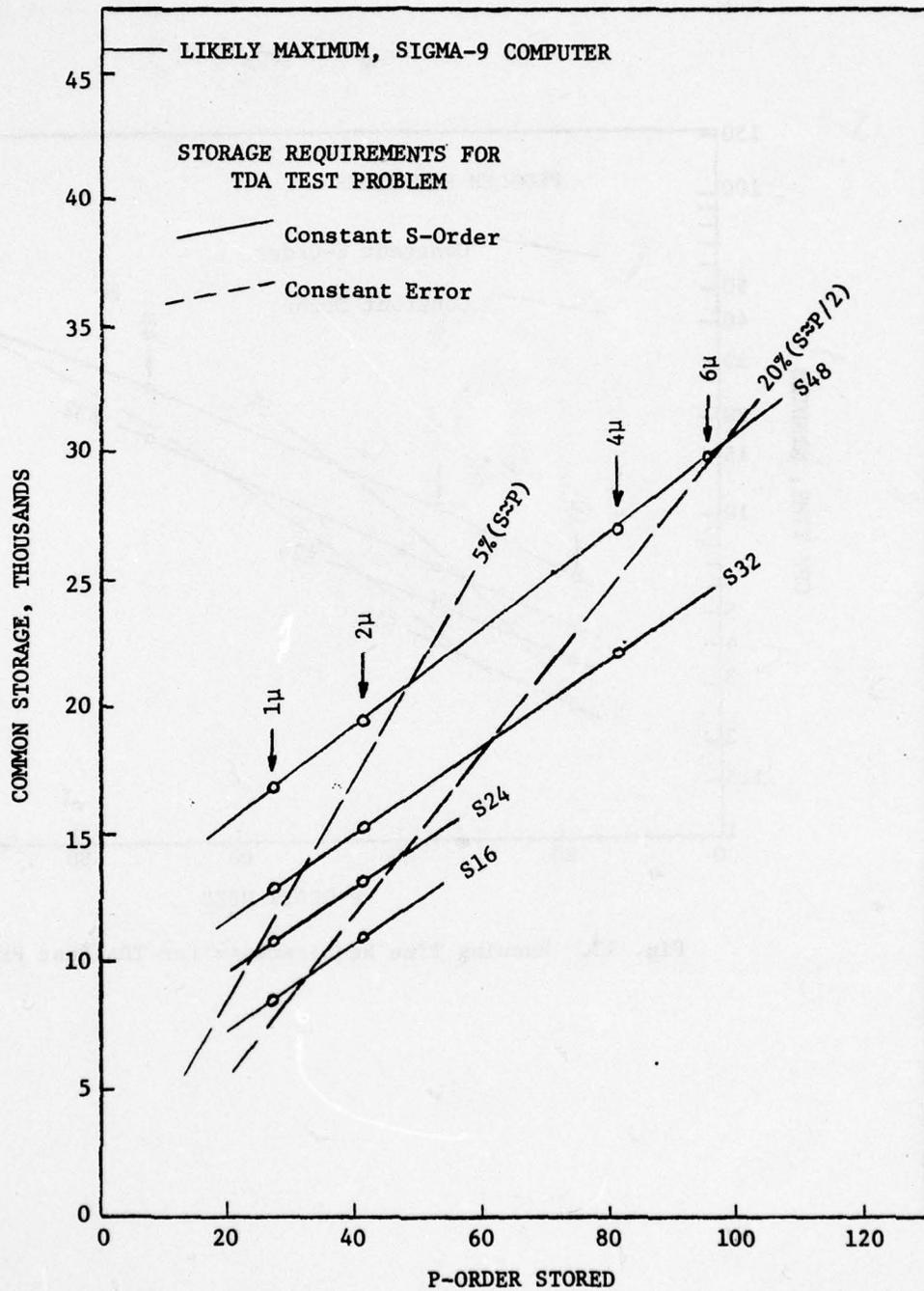


Fig. 12. COMMON Array Storage Requirements for TDA Test Problem

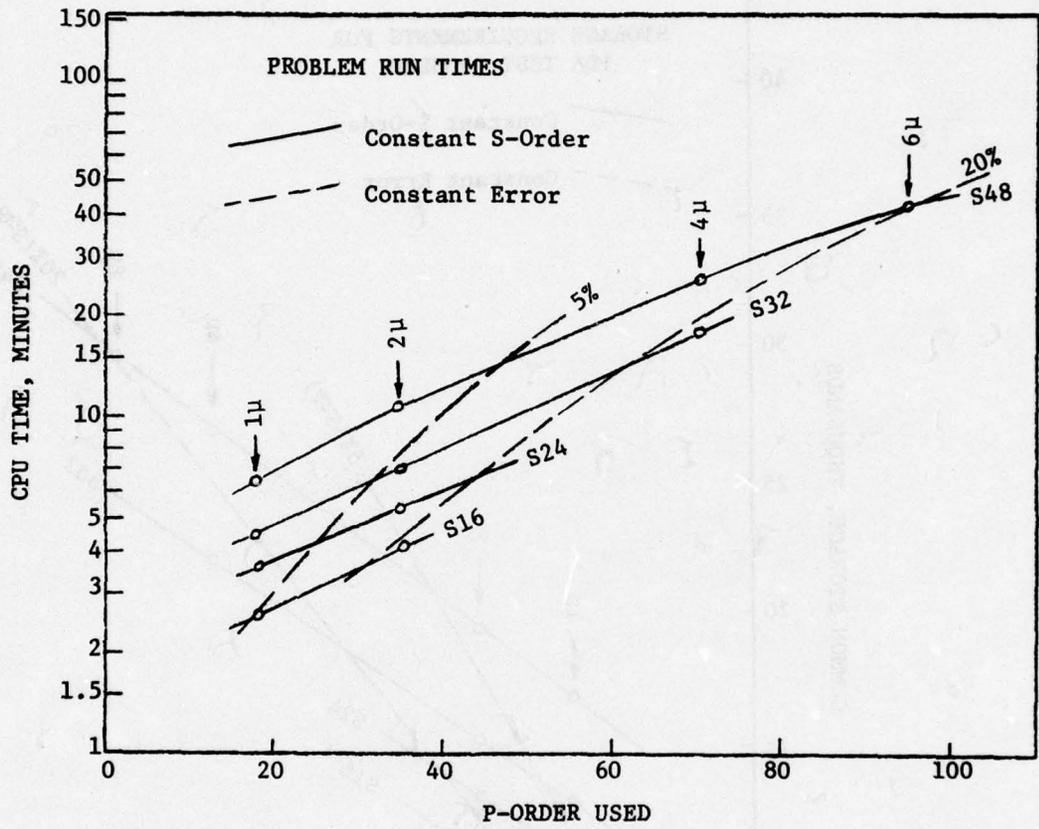


Fig. 13. Running Time Requirements for TDA Test Problem

VIII. TEST EXPERIENCE - CALCULATIONAL RESULTS

Test calculations were performed with the TDA code for 0-1 micron, 0-2 micron, 0-4 micron, and 0-6 micron particle ranges with a wavelength of 0.75 microns, complex refractive index of (1.33,0.0) and a particle size distribution defined by the expression:

$$n(R) = 27R^2 e^{-0.6R} \quad (47)$$

where

R is the particle radius, microns

n(R) is the density distribution in particles cm^{-4} .

Macroscopic Legendre coefficients were generated for each range from individual particle size parameter data stored on the LEG file. The MIELEG code was used to generate the individual particle data and the MACRO code to process it into macroscopic data. Each increase in particle size range requires an increase in the Legendre order for adequate representation of the macroscopic function (phase function integrated over particle size distribution). Also, as more particles are considered, the total cross section* (the first Legendre coefficient) increases. The necessary order and total cross section are shown in Table I.

TABLE I. PARAMETERS OF MACROSCOPIC DATA USED IN TEST CALCULATIONS

Particle Size Range, microns	Maximum Order Req'd ($2\alpha+2$)	Total Cross Section, cm^{-1}
0-1	18	1.72×10^{-6}
0-2	35	5.59×10^{-6}
0-4	70	5.84×10^{-5}
0-6	103	1.72×10^{-4}
0-8	136	3.00×10^{-4}

*Actually, this is the total "scattering" cross section. If a non-zero extinction coefficient were used, the particle density times extinction coefficient would be the "absorption" cross section which, when added to the total scattering cross section, would equal the total cross section.

While orders of 18, 35, and 70 were used for the 1-, 2-, and 4-micron calculations, microscopic particle size data up to only 5.5 microns (but integrated to 6 microns) was used to estimate the cross sections to 6 microns, thereby allowing an order of only 95 to be used. The 8-micron calculation was not performed, due to computer limitations, but was included in the table to note that that rate of increase of cross section decreases when the range of sizes extends well beyond the density distribution peak.

The number of particle sizes considered influences the values of the macroscopic coefficients to some extent. As a test of this, the coefficients of the 0-4 micron range were generated with the MACRO code from LEG file data, using 25 particle sizes, and directly with the MIELEG code using four sizes (PARTS = 3.0). The more approximate MIELEG results included a total cross section of $6.75 \times 10^{-5} \text{ cm}^{-1}$, compared to the MACRO result of 5.84×10^{-5} . The coefficients obtained in these cases are shown in Fig. 14. Whereas the MIELEG-calculated coefficients are greater than the MACRO-calculated coefficients, it is expected that some other choice of the four sizes used by MIELEG might have resulted in smaller coefficients.

It was evident in Section VII that the number of significant coefficients used in an expansion was approximately equal to the number of peaks + valleys in the expanded function (cross section versus angle). Since the peaks and valleys for any single particle size are located at different angles from those for any other particle size, the summation for several particle sizes will result in a larger number of peaks and valleys than there would be for the largest particle size considered. Since the order of the summation is equal to the order of the largest particle size, the function detail will not be completely represented by the summation. Evidence of this insufficiency may be an irregular character of the coefficient distribution, as seen in Fig. 14 for the MACRO-generated coefficients in the 40-60 order range. The expanded function resulting from the MACRO-generated coefficients showed some of the oscillations characteristic of underdetermination

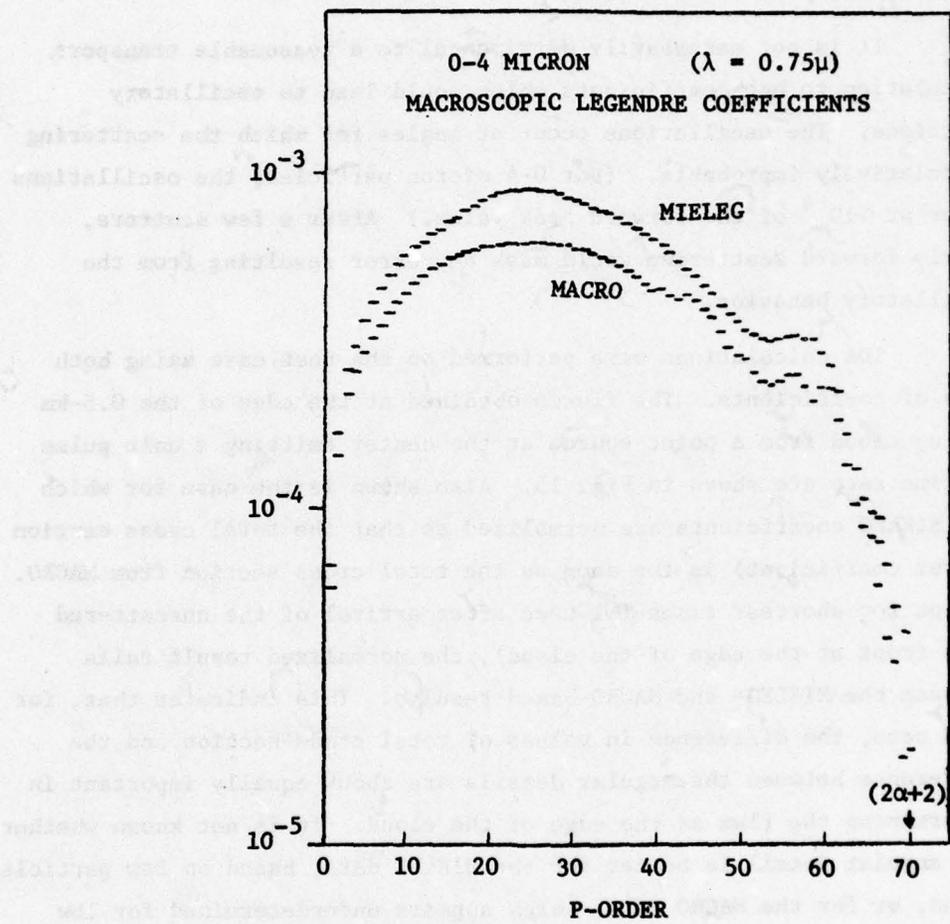


Fig. 14. Comparison of MIELEG- and MACRO-Generated Legendre Coefficients

seen for insufficient order (Fig. 7) or loss of detail by smoothing (Fig. 8). The oscillations were not so apparent in the expanded function resulting from MIELEG-generated coefficients, where fewer particle sizes were considered.

It is not necessarily detrimental to a reasonable transport calculation to use coefficients which would lead to oscillatory functions. The oscillations occur at angles for which the scattering is relatively improbable. (For 0-4 micron particles, the oscillations occur at $\sim 10^{-4}$ of the forward peak value.) After a few scatters, nearly forward scattering would mask any error resulting from the oscillatory behavior.

TDA calculations were performed on the test case using both sets of coefficients. The fluxes obtained at the edge of the 0.5-km radius cloud from a point source at the center emitting a unit pulse at time zero are shown in Fig. 15. Also shown is the case for which the MIELEG coefficients are normalized so that the total cross section (first coefficient) is the same as the total cross section from MACRO. Except for shortest times (~ 1 μ sec after arrival of the unscattered wave front at the edge of the cloud), the normalized result falls between the MIELEG- and MACRO-based results. This indicates that, for this case, the difference in values of total cross-section and the difference between the angular details are about equally important in determining the flux at the edge of the cloud. It is not known whether the angular detail is better for the MIELEG data, based on few particle sizes, or for the MACRO data, which appears underdetermined for low probability scattering. The value of the peak flux is not reliable as insufficient detail was used in spatial increments at the outer edge and time increments when the wave front reaches the outer edge to provide good definition.

The flux at the outer edge of the cloud is plotted versus the time after the unit pulse in Fig. 16 for all particle ranges considered. All results shown here were calculated with $2\alpha+2$ truncation of P-order

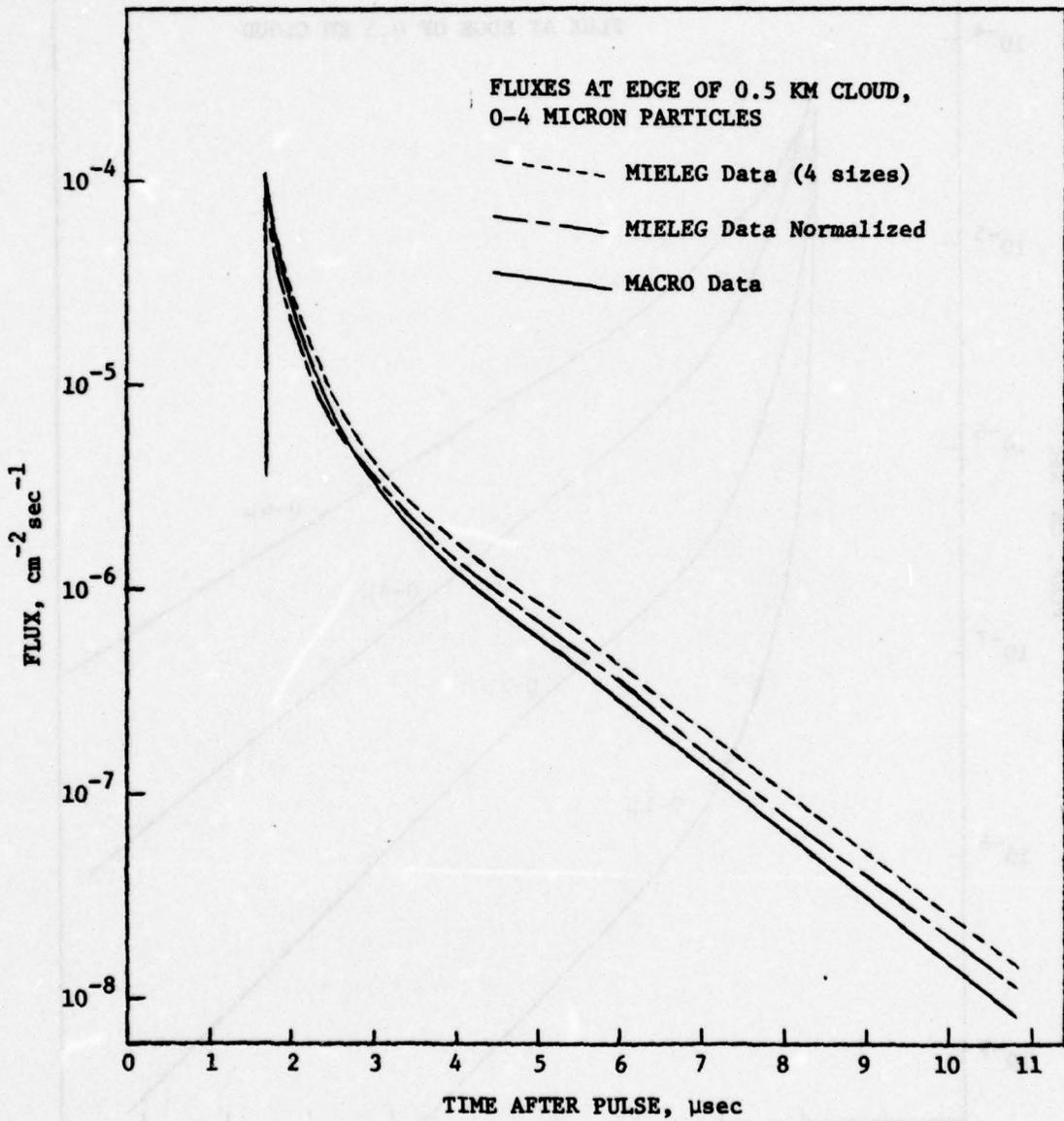


Fig. 15. Effect of Coefficient Data Source on TDA-Calculated Fluxes

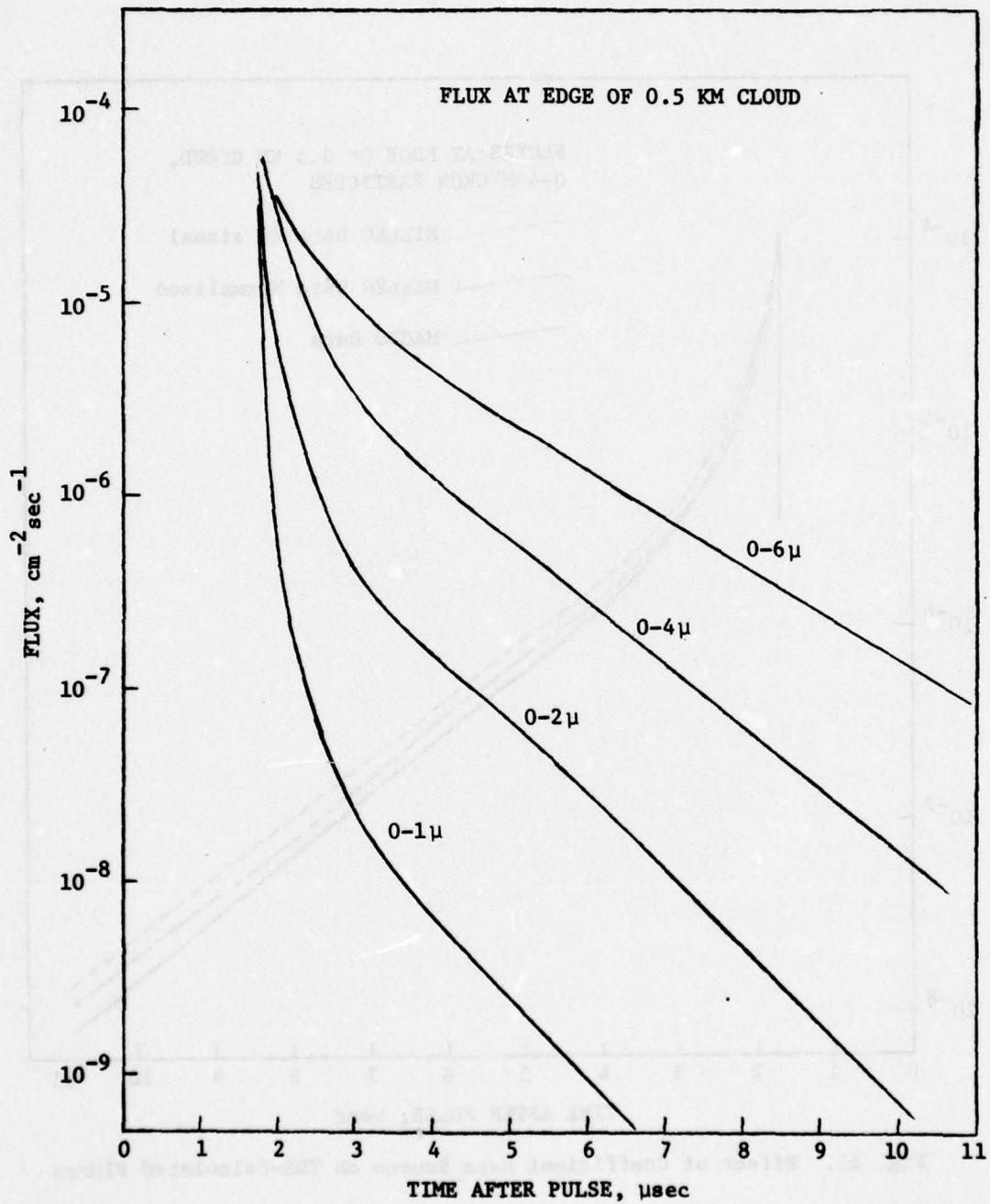


Fig. 16. Effect of Range of Particle Sizes on TDA-Calculated Fluxes

and S48 quadratures. The error in the results from using insufficient S-order is likely to be about 20% for the 0-6 micron case at longest times and less for all other times and size ranges. It is clear that a greater range needs to be considered to observe the effect of the entire range of sizes in the distribution.

With the computer limitations of this study, extension to sufficiently greater sizes would require a significant decrease in P-order or S-order. The possibility of decreasing the S-order by the use of asymmetric quadrature sets has already been mentioned (Section VII). The separation of total cross section and angular detail, as practiced in normalizing MIELEG-calculated coefficients to the total cross section from MACRO, provides a possibility for modest reduction in P-order.

For transport cases of sufficient depth so that the forward direction of the initial pulse becomes greatly spread, a precise definition of forward scattering might not be necessary for adequate results. Less forward scattering, as represented by lower order representation, would predict a longer path to locations of interest, so lower order results might be corrected by multiplying the time scale by a factor equal to the ratio of expected path lengths. As a brief test of this possibility, two low-order angular distributions of cross section were normalized to the total cross sections of two other cases of higher order. The chosen cases are shown in Table II.

TABLE II. TEST CASES FOR LOW-ORDER REPRESENTATION OF HIGH-ORDER SCATTER

Low-Order Cases	High-Order Cases
Rayleigh scatter	0-4 μ (MACRO)
0-2 μ (MACRO)	0-6 μ (MACRO)

Rayleigh scattering represents a low-order limit of light scattering and is given simply by the expression:

$$\sigma(\theta)/\sigma_s = \frac{3}{16\pi} (1 + \cos^2\theta) \quad (48)$$

where $\sigma(\theta)$ is the angular microscopic cross section and σ_s is the total microscopic cross section. Their ratio is the phase function. A Legendre representation of this is obtained by integrating it over all solid angles, weighted by the Legendre polynomial of order n , to obtain the coefficient of order n . Thus,

$$a_n = 2\pi \int_{-1}^1 \frac{\sigma(\theta)}{\sigma_s} P_n(\cos\theta) d(\cos\theta) \quad (49)$$

The first three Legendre polynomials are (using computer indices: 1, 2, 3 for the first three polynomials):

$$\left. \begin{aligned} P_1(\cos\theta) &= 1 \\ P_2(\cos\theta) &= \cos\theta \\ P_3(\cos\theta) &= \frac{3}{2} \cos^2\theta - \frac{1}{2} \end{aligned} \right\} \quad (50)$$

Then $a_1 = 1$, $a_2 = 0$, $a_3 = \frac{1}{2}$. All further terms are zero. That this is an exact representation is confirmed by substituting the coefficients into the expansion:

$$\begin{aligned} \frac{\sigma(\theta)}{\sigma_s} &= \frac{1}{4\pi} \sum_{n=1}^3 a_n P_n(\cos\theta) \\ &= \frac{1}{4\pi} \left[1 \cdot 1 + 0 \cdot \cos\theta + \frac{1}{2} \left(\frac{3}{2} \cos^2\theta - \frac{1}{2} \right) \right] \end{aligned} \quad (51)$$

$$= \frac{1}{4\pi} \left[2 + \frac{3}{4} \cos^2 \theta - \frac{1}{4} \right] = \frac{3}{16\pi} \left[1 + \cos^2 \theta \right],$$

the original expression. A Rayleigh scattering angular distribution can, therefore, be used to represent a higher order case by a P3 data (P2 in traditional indexing), where the first coefficient is the total macroscopic cross section, the second is zero, and the third is one-half of the total cross section. This was done for both 0-4 micron and 0-6 micron cases. In addition, the 0-2 micron P35 angular distribution was normalized to both the 0-4 micron and 0-6 micron cases, using the MACEDIT code. The results of both representations of the 0-4 micron case, along with the original 0-4 micron results, are shown in Fig. 17. Similarly, results for the 0-6 micron case are shown in Fig. 18.

In Fig. 17, both normalized cases have initial peaks, probably from local scatter of the passing wave front, followed by multi-scatter contributions. The Rayleigh scatter case, however, is depressed for times immediately after the passage of the front, lacking a large component of strongly forward-scattered radiation. Thereafter, it has a very strong delayed component from other scatters. The case normalized from the 0-2 micron distribution is intermediate to the two others, having more forward-scattering contribution than the Rayleigh and less than the true 0-4 micron case. Within about 4 μ sec of the pulse, the flux then appears of the form of the true case, but delayed by longer scattering paths.

In Fig. 18, the Rayleigh scatter case shows no initial peak at all, indicating that the cross section is large enough so that the wave front is very weak and local scattering from it near the edge of the cloud is small. All Rayleigh scatter contributions are delayed by relatively long multiple-scattering paths. The case normalized from 0-2 microns is similar in shape to the true 0-6 micron case, but delayed. This shows some promise that the technique of using a normalized lower-order cross section set and an adjusted time scale might give reasonable approximation to the true higher-order case for large numbers of scatters.

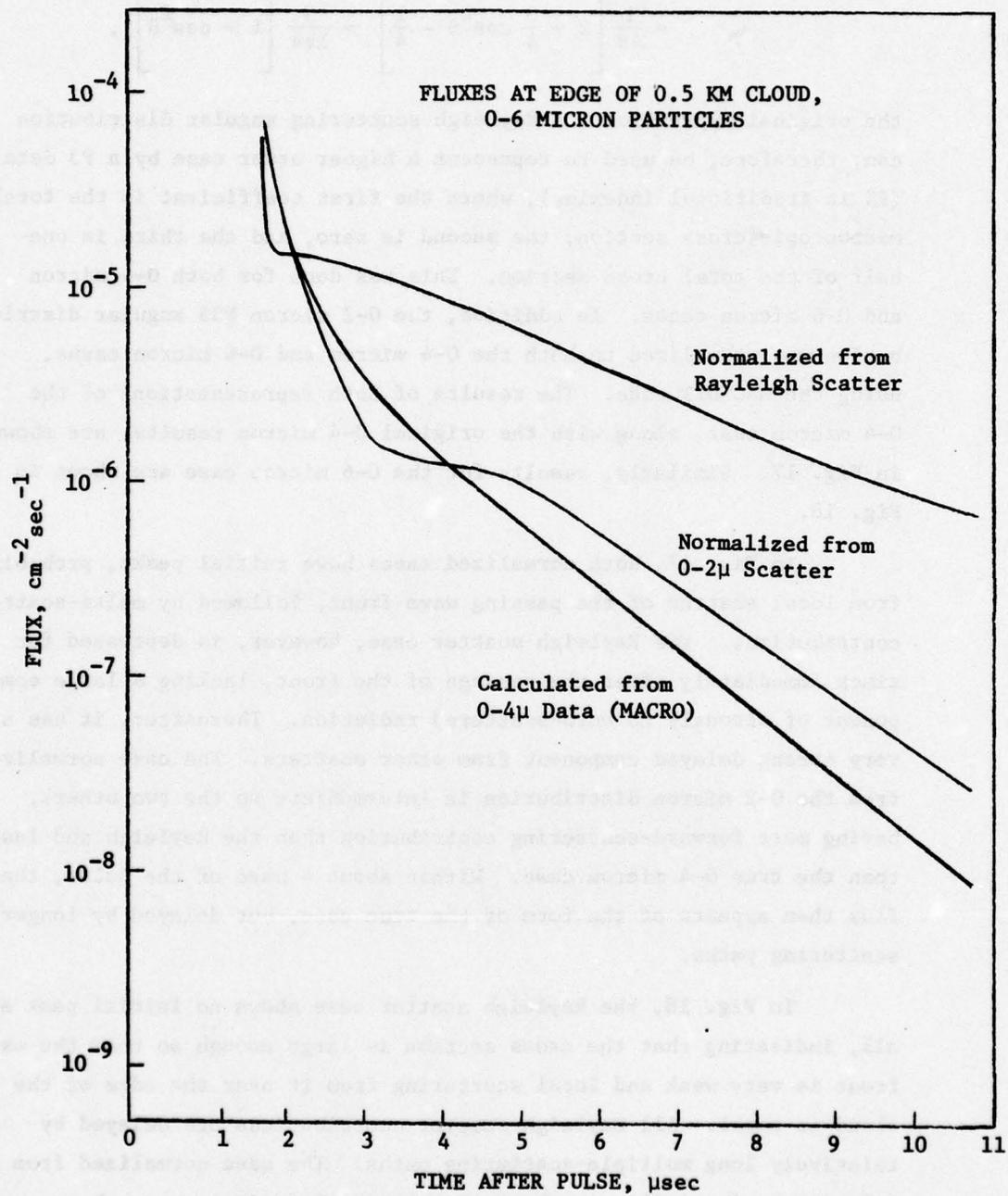


Fig. 17. Comparison of Cross-Section Normalizations, 0-4 Micron Particles

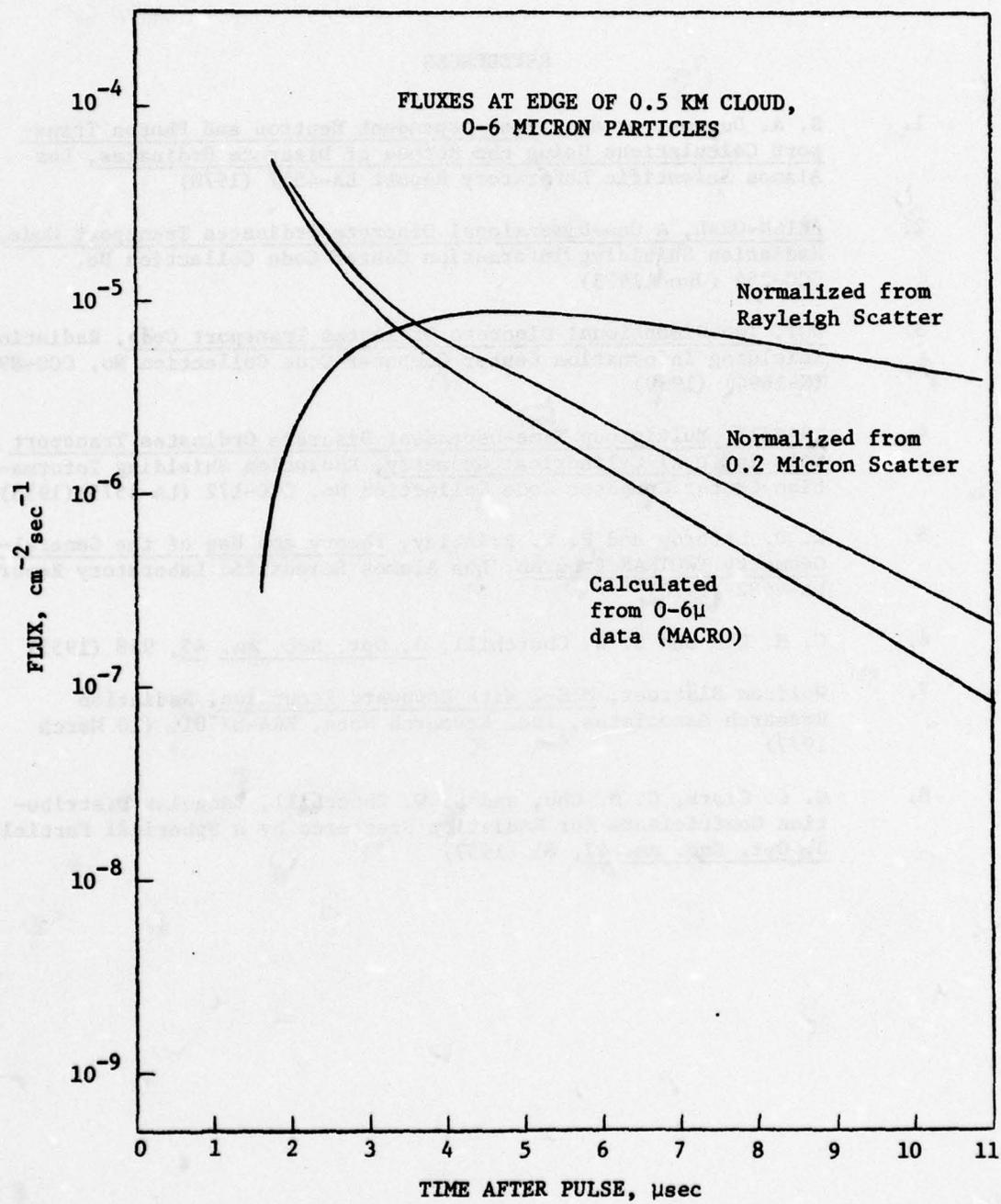


Fig. 18. Comparison of Cross-Section Normalizations, 0-6 Micron Particles

REFERENCES

1. S. A. Dupree, et al., Time-Dependent Neutron and Photon Transport Calculations Using the Method of Discrete Ordinates, Los Alamos Scientific Laboratory Report LA-4557 (1970)
2. ANISN-ORNL, A One-Dimensional Discrete Ordinates Transport Code, Radiation Shielding Information Center Code Collection No. CCC-254 (June 1973)
3. DOT, Two-Dimensional Discrete Ordinates Transport Code, Radiation Shielding Information Center Computer Code Collection No. CCC-89 (K-1694) (1969)
4. TRANZIT, Multigroup Time-Dependent Discrete Ordinates Transport Code in (ρ, z) Cylindrical Geometry, Radiation Shielding Information Center Computer Code Collection No. CCC-172 (LA-4575) (1973)
5. K. D. Lathrop and F. W. Brinkley, Theory and Use of the General-Geometry TWOTRAN Program, Los Alamos Scientific Laboratory Report LA-4432 (1970)
6. C. M. Chu and S. W. Churchill, J. Opt. Soc. Am. **45**, 958 (1955)
7. Wolfram Blättner, MIE-2 With Downward Recursion, Radiation Research Associates, Inc. Research Note, RRA-N7701, (10 March 1977)
8. G. C. Clark, C. M. Chu, and S. W. Churchill, "Angular Distribution Coefficients for Radiation Scattered by a Spherical Particle," J. Opt. Soc. Am. **47**, 81 (1957)

APPENDIX A. MIELEG UTILIZATION INSTRUCTIONS

The MIELEG code calculates Legendre coefficients of interaction probability per unit particle density (microscopic cross-section coefficients, a_n) or interaction probability per unit path (macroscopic cross-section coefficients, A_n). Input data are particle size range and distribution, radiation wavelength, and complex refractive index. It is based on the MIE-2 code and direct evaluation of Legendre coefficients (Refs. 7 and 8). It produces a file, LEG, of microscopic coefficients and a file, MAC, of macroscopic coefficients.

A.1 MIELEG Input Description

RMIN, RMAX, PARTS, PM (Complex), WAVEL, NEQ (6F10.0, I5)

- RMIN - Minimum particle radius considered, microns
- RMAX - Maximum particle radius considered, microns
- PARTS - Number of intervals in range of considered radii.
If 0.0 used, only RMAX considered, no size distribution necessary, no MAC file produced.
- PM - Complex refractive index, β , composed of two parts:
 - a) M, real index of refraction relative to the surrounding material, and
 - b) K, extinction coefficient of the particle material.
- WAVEL - Wavelength of the radiation, microns
- NEQ - Number of equation describing particle size distribution, 0 or 1
 - 0: $N(R) = P1 * R^{**} P2 * EXP(-P3 * R^{**} P4)$
 - 1: $N(R) = P1$ for $R \leq P2$
 $N(R) = P3 * R^{**} P4$ for $R > P2$

P1, P2, P3, P4 (4F10.0) (Ignored if PARTS = 0.0)

Parameters described by choice of NEQ, on previous input record.

Output Options 4(LX,A4) (Ignored if PARTS = 0.0)

Possible options:

XSOR or XSLA, TRUN, AGIN, SMOO

Output options are entered in Hollerith (alphabetic) form. They can be in any order, but must be spaced as indicated by the LX. They can be conveniently input with a leading space, then separated by commas.

XSOR or XSLA - MAC option. Most codes requiring Legendre coefficient cross-section data are of the Oak Ridge (FIDO) type or Los Alamos (DTF) type. The two generally differ by a factor of $(2N+1)$ where N is the order (initial order: zero) of the $N+1$ st coefficient. For a discussion of the evaluation of coefficients from Mie theory and the type choice, see Section IV: Conversion of Mie Parameters to Legendre Coefficient Data.

Option:	XSOR	XSLA
Type:	Oak Ridge	Los Alamos
Typical Codes:	ANISN, TDA	DTF
	DOT	TWOTRAN
	MORSE	TRANZIT

TRUN - Truncation option.

The MIELEG code produces coefficients for a higher order (more values) than is normally necessary. The MIELEG code uses all of these coefficients to calculate the macroscopic cross-section of equation (3) (labeled F on the printed output) for 0 through 180 degrees. It then truncates the series of coefficients and recalculates the cross-section values. Truncations are chosen according to several criteria:

- 1) where the coefficients is first smaller than the initial coefficient
- 2) where N is first greater than $2\alpha+2$
- 3) at local minima in the coefficient sequence.

The cross sections resulting from each of these truncations can be inspected to find an adequate order. Truncations that are too severe often result in a few negative cross-section values. Truncation at $2\alpha+2$ is generally adequate.

AGIN - Refitting option.

Test of the consistency of the Legendre polynomials and expansions. This option integrated the expanded cross sections over all angles, weighted by Legendre polynomials to recalculate the coefficients. In general, the values agree with the originals to about 3-place accuracy until $N\sqrt{2}\alpha$, after which the original coefficients have contributed little to the cross sections, so cannot be extracted from them.

SMOO - Smoothing option.

This option was added to refit cross-section data using smoothed values for data at angles greater than those in the large forward-scattering peak. Tests with this option did not result in significantly different coefficient sets, indicating that the Legendre order is determined by the forward-scatter peak.

THRESHF, WIDTH (2E10.0)

Smoothing parameters. If the SMOO option is used, those parameters are used. Cross sections are averaged over WIDTH number of degrees for all angles greater than the angle for which the cross-section value falls to $THRESH * F(1)$ where $F(1)$ is the cross-section value at zero degrees (the maximum of the forward peak).

A.2 MIELEG Output Description - Printout

The MIELEG code prints out all input data, then radii and relative densities if more than one calculated. For each radius, the size parameter (ALPHA), cross-section ratio $K(\alpha, \beta)$ (SCKRO), and Legendre order (LESS) are printed, followed by the actual Legendre coefficients.

If several sizes are considered, the macroscopic coefficients are printed and these are expanded into the angular dependent cross sections, which are also printed.

If the TRUN option is used, the expansion is printed for each truncation. If the SMOO and AGIN options are used, the resulting coefficients are printed along with the full expansion. The TRUN option, along with SMOO or AGIN also causes the truncated expansions for these cases to be printed.

A.3 MIELEG Output Description - Files

The MIELEG code produces two files, LEG, containing coefficients for single particle sizes, and MAC, with coefficients integrated with particle size distribution. Other files may be produced by a MIELEG run. In case a run is interrupted, the LEG file existing before the run would be found on BACKLEG. The new, partially completed file, might be found on NEWLEG. If the run string were completed, and included a run of the LEGEDIT code, any coefficients created which had a different complex refractive index from those already on the LEG file would be written on the OTHERLEG file.

A.4 MIELEG FORTRAN Listing

ROUTINE MIELEG - NEW SYSTEM, REGRESSION SCHEME
CODE TO FIT THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

DIMENSIONALITY

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

MIELEG CODE

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

CONSTRUCTION OF THE PHASE FUNCTIONS WITH ERROR FOR YINDELIS
COMPLEX ALGEBRA (R-1987-021, R-1987-022, R-1987-023, R-1987-024, R-1987-025)

```

C
C ROUTINE MIELEG - NEW (7/7/77) RECURSION SCHEME
C
C CODE TO FIT MIE PHASE FUNCTIONS WITH LEGENDRE POLYNOMIALS
C COMPLEX A(500),B(500),PSIR(500),PSIRC(500),BCA(2),ACAPB,
1 BETA1,PM,PSRC,ZETA
C
C DIMENSION PSI(3),CHI(3),COEF(500),
1 RI(200),ENR(200)
COMMON ALPHA,PII
COMMON/POUT/RMIN,RMAX,PM,WAVEL,NEQ,P1,P2,P3,P4,NRS,
1 LESS,ENTOT,COEFXS(500)
DIMENSION SUMW(500),SUMV(500)
C
C EQUIVALENCE(A(1),PSIR(1)),(B(1),PSIRC(1))
G(J,K) = (REAL(A(K))*REAL(A(J)) + AIMAG(A(K))*AIMAG(A(J)) +
1 REAL(B(K))*REAL(B(J)) + AIMAG(B(K))*AIMAG(B(J)))
2 *(2.*J+1.)/(J*(J+1.))* (2.*K+1.)/(K*(K+1.))
F(J,K) = (REAL(A(K))*REAL(B(J)) + AIMAG(A(K))*AIMAG(B(J)) +
1 REAL(B(K))*REAL(A(J)) + AIMAG(B(K))*AIMAG(A(J)))
2 *(2.*J+1.)/(J*(J+1.))* (2.*K+1.)/(K*(K+1.))
C
C G(K,J) = (2*N+1)*(J*(J+1)+K*(K+1)-N*(N+1))*2/4.
C
C REWIND 11
REWIND 12
PII = 4.0 * ATAN(1.0)
PI2 = 2.0 * PII
C
C DO 2 I=1,500
COEFXS(I) = 0.0
2 CONTINUE
C
C READ (5,600) RMIN, RMAX, PARTS, PM, WAVEL, NEQ
WRITE(6,610) RMIN, RMAX, PARTS, PM, WAVEL, NEQ
NRS = PARTS + 1.0
IF (PARTS.GT.0.0) GO TO 25
RI(1)=RMAX
GO TO 72
C
C USE PARTS.EQ.0.0 TO GET COEFS FOR SINGLE SIZE ONLY.
C TRUNCATING, REFITTING TESTS SUPPRESSED. RMIN IGNORED.
C NO SIZE DISTRIBUTION DATA REQUIRED.
C
C
C 25 DELR = 2.0*(RMAX-RMIN)/(PARTS*(PARTS+1.0))
C
C *** CALCULATE PARTICLE RADII ***
C
C RI(1) = RMIN
C
C DO 28 I=2,NRS
I1 = I - 1
28 RI(I) = RI(I1) + I1 *DELR
C
C *** CALCULATE PARTICLE SIZE DISTRIBUTION ***
C
C IF(NEQ)20,32,40
32 READ(5,620) P1,P2,P3,P4

```

```

WRITE(6,630) P1,P2,P3,P4
DO 37 I=1,NRS
37 ENR(I)=P1*RI(I)**P2*EXP(-P3*RI(I)**P4)
GO TO 70
40 READ(5,620) P1,P2,P3,P4
WRITE(6,640)P1,P2,P3,P4
C
DO 18 I=1,NRS
IF(RI(I).GT.P2) GO TO 57
ENR(I) = P1
GO TO 18
57 ENR(I) =P3*RI(I)**P4
18 CONTINUE
C
70 WRITE(6,650) NRS
WRITE(6,661)(RI(I),ENR(I),I=1,NRS)
C
72 ENTOT=0.0
CEXT =0.0
SCOF =0.0
C
DO 500 KC=1,NRS
C
C      *** CALCULATE MIE VARIABLES ***
C
EXKRO = 0.0
SCKRO = 0.0
ALPH = WAVEL/PI2
71 X = RI(KC)/ALPH
C
C TEST TO AVOID SINGULARITY AT MULTIPLES OF PI
C
IF(ABS(SIN(X)).GT.0.001 ) GO TO 73
C
RI(KC)=RI(KC)*1.001
C
WRITE(6,74)KC
74 FORMAT(/,'R NUMBER',I3,' FUDGED TO AVOID SINGULARITY '
1 'AT MULT. OF PI')
C
GO TO /1
C
C CHANGE IN X TO AVOID ODD RESULT AT SINGULARITY.
C
73 XYZ = X
ALPHA=X
WOP = 1.0E-08 *ALPH**2
COFPH=2./(X*X)
C
IF(X.LT.100.) GO TO 10
L =X*CABS(PM)
K1 =1.1*L
GO TO 11
10 L=15.0*SQRT(1.+0.01*X*X)
K1 = 1.5 * L
11 K2 = K1-1
IF(L.LT.501) GO TO 12

```

```

WRITE(6,670) L
STOP
12 XI =1.0/X
   BETAI =1.0/(PM*X)
   PSR = 0.0
   PSRC = CMPLX(0.0,0.0)
   DO 15 I=1,K2
   NR=K1-I
   IA =2*NR+1
   PSR =1.0/(IA*XI-PSR)
   PSRC=1.0/(IA*BETAI-PSRC)
   IF(NR.GT.L) GO TO 15
   PSIR(NR) = CMPLX(PSR,0.0)
   PSIRC(NR) =PSRC
15 CONTINUE
C
   CHIO = COS(X)
   PSIO = SIN(X)
C
   DO 50 I=1,L
C
C   *** CALCULATE MIF SERIES COEFFICIENTS ***
C
   PSR = REAL(PSIR(I))
   I1 = I + 1
   IF(I.GT.2) GO TO 30
   IF(I.EQ.2) GO TO 20
   PSI(1)=PSR*PSIO
   PSI(2)=REAL(PSIR(2))*PSI(1)
   CHI(1)=XI*CHIO + PSIO
   CHI(2) = 3.0*XI*CHI(1)-CHIO
   BCA(1)= -XI + 1.0/(XI-CMPLX(0.0,-1.0))
20 PSI(3)=PSI(1)
   CHI(3)= CHI(1)
   GO TO 35
30 PSI(3)=PSR*PSI(2)
   CHI(3) = (2*I-1)*XI*CHI(2)-CHI(1)
35 ZETA = CMPLX(PSI(3),CHI(3))
   ACAPA = 1.0/PSR - I*XI
   ACAPB = 1.0/PSIRC(I)-I*BETAI
   BCA(2)=-I1*XI + 1.0/(I1*XI-BCA(1))
   A(I) = PSI(3)*(ACAPB-PM*ACAPA) / (ZETA*(ACAPB -PM*BCA(1)))
   B(I) = PSI(3)*(ACAPA-PM*ACAPB) / (ZETA*(BCA(1)-PM*ACAPB ))
   BCA(1) = BCA(2)
   ICONST = 2*I +1
   EXKRO =ICONST * REAL(A(I)+B(I)) + EXKRO
   SCKRO = ICONST *(CABS(A(I)**2)+CABS(B(I)**2)) + SCKRO
   IF(I.LT.3) GO TO 50
   PSI(1) =PSI(2)
   PSI(2) = PSI(3)
   CHI(1) = CHI(2)
   CHI(2) = CHI(3)
50 CONTINUE
C
   EXKRO = EXKRO * COEPH
   SCKRO = SCKRO * COEPH
C
   FUD = (2.0/(X**2 * SCKRO ))
   LAST = L

```

```

C
C   LMAX = L * (L+1)/2
C
C   L = LENGTH OF MIE-2 SERIES ON A AND B
C
C   *** CALCULATE LENGTH OF LEGENDRE SERIES ***
C
C   IF(X.LE.15.0) GO TO 49
C   LESS = 21.0*EXP(0.040611 * X)
C   GO TO 51
49  LESS = 2.0 * X + 11.0
C
C   51 IF(LESS.GT.2*L) LESS = 2*L
C
C   THIS IS THE BEGINNING OF THE NEW RECURSION SCHEME
C   (D. COLLINS, RRA, 7/7/77)
C
C   MAX = LESS + 1
C   DO 52 N=1,MAX
52  SUMW(N) = 0.0
C   DO 100 K=1, LAST
C   Y = K
C   DO 100 J=K, LAST
C   X = J
C   BWKJ = G(K,J)
C   N=J-K
C   CKJ = C(K,J)*2.0/(1.0+IFIX(K/J))
C   IF (J.EQ.K) WJKO = 1.0/(2*Y+1)
C   IF (J.GT.K) WJKO = WJKO*X*(2*X-2*Y-1)/((2*X+1)*(X-Y))
C   WB4 = WJKO
C   M = J-K+1
C   SUMW(M) = BWKJ*WB4*CKJ+SUMW(M)
C   NL = J-K+3
C   NU = J+K+1
C   TOO=2.0/(1.0+IFIX(K/J))
C   DO 100 M=NL,NU,2
C   N=M-1
C   Z = M-1
C   WB4 = WB4*(X+Z-Y-1)*(Y+Z-X-1)*(X+Y+Z)*(X+Y-Z+2)/
8   ((X+Y-Z+1)*(X+Y+Z+1)*(X+Z-Y)*(Y+Z-X))
C   SUMW(M) = SUMW(M) + BWKJ*WB4*C(K,J)*TOO
100 CONTINUE
C   DO 120 N=1,MAX
120 SUMV(N) = 0.0
C   DO 200 K=1, LAST
C   Y = K
C   DO 200 J=K, LAST
C   X = J
C   BVKJ = F(K,J)
C   IF (J.EQ.K) VJKO = 3*Y*(Y+1)/(2*Y+1)
C   IF (J.GT.K) VJKO = VJKO*(2*X-2*Y+3)*(X+1)/((X-Y)*(2*X+1))
C   VB4 = VJKO
C   N = J-K+2
C   SUMV(N) = SUMV(N)+BVKJ*VB4*2.0/(1.0+IFIX(K/J))
C   IF (K.EQ.1) GO TO 200
C   NL = J-K+4
C   NU = J+K
C   TOO=2.0/(1.0+IFIX(K/J))

```

```

      DO 190 M=NL,NU,2
      Z = M-1
      VB4 = VB4*(2*Z+1)*(X+Z-Y)*(Y+Z-X)*(X+Y+Z+1)*(X+Y-Z+1)/
      & ((2*Z-3)*(X+Z-Y-1)*(Y+Z-X-1)*(X+Y+Z)*(X+Y-Z+2))
      SUMV(M) = SUMV(M) + BVKJ*VB4*T00
190  CONTINUE
200  CONTINUE
      DO 250 N=1,MAX
250  COEF(N) = FUD*(SUMW(N)+SUMV(N))
C
C   THIS IS THE END OF THE NEW RECURSION SCHEME
C
      IF(PARTS.EQ.0.0) GO TO 402
C
C   *** FOLD LEGENDRE COEFFICIENTS WITH PARTICLE DISTRIBUTION ***
C
      IF(KC.GT.1.AND.KC.LT.NRS) GO TO 310
      IF(KC.EQ.NRS) DELR=(RI(KC)-RI(KC-1))/2.0
      IF(KC.EQ.1) DELR=(RI(KC+1)-RI(KC))/2.0
      GO TO 330
310  DELR=(RI(KC+1)-RI(KC-1))/2.0
330  EDRR =ENR(KC)*DELR*RI(KC)*RI(KC)
      CEXT = CEXT + EXKRO*EDRR
      SCOE = SCOE +SCKRO * EDRR
C
C
C
402  WRITE(6,680) PM,XYZ,SCKRO,LESS
      WRITE(6,662) (COEF(N),N=1,LESS)
C
C   PUT COEFFICIENTS ON LIBRARY (TEMP ON NEWLEG, COPY TO LEG)
C
      WRITE(12)XYZ,PM,SCKRO,LESS
      WRITE(12)(COEF(N),N=1,LESS)
      IF (PARTS.EQ.0.0) GO TO 570
C
C
      DO 450 MM=1,LESS
      COEFXS(MM)=COEFXS(MM)+COEF(MM)*SCKRO*PII*RI(KC)*RI(KC)
      I *ENR(KC)*DELR*1.0E-08
450  CONTINUE
      ENTOT=ENTOT + ENR(KC)*DELR
500  CONTINUE
C
C
      WRITE(6,720) ENTOT, LESS
      WRITE(6,555)(COEFXS(N),N=1,LESS)
555  FORMAT(/,'MACROSCOPIC X-S COEFFICIENTS',/,(1P6E15.6))
C
      CEXT = PII*CEXT*1.0E-08
      SCOE = PII * SCOE * 1.0E-08
      ABOE=CEXT-SCOE
C
      WRITE(6,710) CEXT,SCOE,ABOE
C
C
C   COPY OLD LIBRARY
C

```

```

570 L=0
580 READ(11,END=590) XYZ,PM,SCKRO,LESS2
    READ(11)(COEF(N),N=1,LESS2)
    WRITE(12)XYZ,PM,SCKRO,LESS2
    WRITE(12)(COEF(N),N=1,LESS2)
    L=L+1
    GO TO 580
590 REWIND 11
    RFWIND 12
    WRITE(6,595) L
595 FORMAT(/,'READ',I3,' SETS OF COEFFICIENTS FROM LIBRARY '
1 'FOR UPDATE.')
C
    IF(PARTS.EQ.0.0)STOP
C
C   OUTPUT OPTIONS
C
599 CALL OUTOP
C
600 FORMAT(6F10.0,I5)
610 FORMAT(/,2X,'INPUT DATA',//,2X,'RMIN = ',F10.2,2X,'MICRONS',
1/,2X,'RMAX = ',F10.2,2X,'MICRONS',/2X,'PARTS = ',F10.2,/,
22X,'PM = ',2E15.6,/,2X,'WAVEL = ',F10.2,2X,'MICRONS',
3/,2X,'NEQ = ',I5)
620 FORMAT(4F10.0)
630 FORMAT(/,2X,'P1 = ',1PE15.6,/,2X,'P2   = ',1PE15.6,/,2X,
1'P3 = ',1PE15.6,/,2X,'P4   = ',1PE15.6)
640 FORMAT(/,2X,'P1   = ',1PE15.6,/,2X,'P2   = ',1PE15.6,/,
12X,'P3 = ',1PE15.6,/,2X,'P4   = ',1PE15.6)
650 FORMAT(/,2X,'NRS = ',I5,/)
661 FORMAT(/,3X,'RADIUS',6X,'DISTRIB. DENS.'/(1P2E15.5))
662 FORMAT(/,'LEGENDRE COEFFICIENTS. SINGLE SIZE'/(1P6E15.6))
670 FORMAT(/,2X,'DIMENSIONS TOO SMALL. L = ',I5)
680 FORMAT(/,2X,'PM = ',2E15.6,2X,'ALPHA = ',F15.2,/,
12X,'SCKRO = ',1PE15.6,2X,'LESS = ',I5,/)
700 FORMAT(I10,5X,1P2E15.7)
710 FORMAT(/,2X,'MACROSCOPIC EXTINCTION CROSS SECTION = ',1PE12.5,
1    /,2X,'MACROSCOPIC SCATTERING CROSS SECTION = ',E12.5,
2    /,2X,'MACROSCOPIC ABSORPTION CROSS SECTION = ',E12.5,/)
720 FORMAT(/,2X,'AVERAGE LEGENDRE COEFFICIENTS',/,
1    2X,'ENTOT = ',1PE15.6,2X,'LESS = ',I5,/)
C
    STOP
    END
    SUBROUTINE OUTOP
    COMMON/POUT/RMIN,RMAX,PM,WAVEL,NEQ,P1,P2,P3,P4,NRS,LESS,
1 EN,COEFXS(500)
    DATA IA0,IA1,IA2,IA3,IA4/4HXSOR,4HXSLA,4HSMO0,4HAGIN,4HTRUN/
    DATA I1,I2,I3,I4/4*0/
    COMMON ALPHA
    COMMON/OUT/I1,I2,I3,I4
    COMPLEX PM
C
C   OTHER OUTPUT
C
    READ(5,302) IOUT1,IOUT2,IOUT3,IOUT4
302 FORMAT(4(1X,A4))
C

```

```

C CHECK HOLLERITH VALUES
C
  WRITE(6,303)IOUT1,IOUT2,IOUT3,IOUT4,IA0,IA1,IA2,IA3,IA4
303 FORMAT(/,'OPTIONS CHOSEN: ',4(1X,A4),/, 'AVAILABLE OPTIONS: ',
  1 5(1X,A4))
CCCCCCCCCCCCCCCCCCCCCCCC
C
C I1=1 IF X-S SET FOR ANISN (OR OTHER OAK RIDGE CODE) PRODUCED
  IF(IOUT1.EQ.1A0.OR.IOUT2.EQ.1A0.OR.IOUT3.EQ.1A0.
  1 OR.IOUT4.EQ.1A0) I1=1
  I1WAS=I1
C I1=2 IF X-S SET FOR DTF (OR OTHER LOS ALAMOS CODE) PRODUCED
  IF(IOUT1.EQ.1A1.OR.IOUT2.EQ.1A1.OR.IOUT3.EQ.1A1.
  1 OR.IOUT4.EQ.1A1) I1=2
C I2=1 IF SMOOTHING TESTED
  IF(IOUT1.EQ.1A2.OR.IOUT2.EQ.1A2.OR.IOUT3.EQ.1A2.
  1 OR.IOUT4.EQ.1A2) I2=1
C I3=1 IF REFITTING OF COEFS TO F TESTED
  IF(IOUT1.EQ.1A3.OR.IOUT2.EQ.1A3.OR.IOUT3.EQ.1A3.
  1 OR.IOUT4.EQ.1A3) I3=1
C I4=1 IF TRUNCATION TESTED
  IF(IOUT1.EQ.1A4.OR.IOUT2.EQ.1A4.OR.IOUT3.EQ.1A4.
  1 OR.IOUT4.EQ.1A4) I4=1
C
  IF(I1.EQ.0) GO TO 288
  IF(I1WAS.EQ.1.AND.I1.EQ.2) GO TO 270
  GO TO 288
270 I1=1
  WRITE(6,272)
272 FORMAT(/,'CANNOT PRODUCE O.R. AND L.A. FORMAT XS '
  1 'SETS IN THE SAME RUN. O.R. (E.G., ANISN) CHOSEN.')
287 FORMAT(' OUTPUT INDICES: ',15,5X,315)
289 FORMAT(' OUTPUT INDICES: ',5X,415)
288 IF(I1.EQ.0) GO TO 305
  IF(I1.EQ.1) GO TO 285
  DO 280 MM=1,LESS
  COEFXS(MM)=COEFXS(MM)/(2.*FLOAT(MM)+1.)
280 CONTINUE
  WRITE(6,289)I1,I2,I3,I4
  GO TO 281
285 WRITE(6,287)I1,I2,I3,I4
281 ZERO=0.0
  MORDER=2.*ALPHA+3.
  WRITE(15,4)
  4 FORMAT(/,'MAC DATA PRODUCED BY MIELEG')
  WRITE(15,291)RMIN,RMAX,PM,WAVEL,NEQ,P1,P2,P3,P4,NRS,LESS.EN,MORDER
291 FORMAT(/,'PARAMETERS',/,5E15.6,/,14,4E15.6,/,215,E15.6,15)
  WRITE(15,292)ZERO,ZERO,COEFXS(1),COEFXS(1),
  1 (ZERO,ZERO,ZERO,COEFXS(MM),MM=2,LESS)
292 FORMAT(/,'6E12.5 FORMAT CROSS-SECTIONS',/, '14**',/2F12.1,
  1 2E12.5,/, (3F12.1,E12.5))
  IF(I1.EQ.1)WRITE(6,293)
293 FORMAT(/,'MACR X-S DATA WRITTEN ON MAC FILE (15).- OPTION XSOR')
  IF(I1.EQ.2) WRITE(6,294)
294 FORMAT(/,'MACR X-S DATA WRITTEN ON MAC FILE(15). - OPTION XSLA')
C
305 CALL TESTAN(0,COEFXS,LESS)
  RETURN
  FND

```

```

SUBROUTINE TESTAN(I,AN,LESS)
  DIMENSION X(181),AN(500),AN2(500),AN3(500),P(500),
1 P2(500),P3(181),F(181),F2(181),F3(181),N(181),TW(181),
2 FFI(181)
  COMMON ALPHA,PI,IP,IP2,IP3
  COMMON/OUT/I1,I2,I3,I4
C
  IF(LESS.LT.501) GO TO 10
  WRITE(6,5) LESS
5 FORMAT(/,'LESS = ',I4,' MUST CHANGE DIMENSION IN TESTAN AND LEGCHK
1')
  STOP
10 IF(I.GT.1) GO TO 42
  T1=PI/180.
  DO 30 J=1,181
  T=T1*(J-1)
  X(J)=COS(T)
30 CONTINUE
C
C PRODUCE LEG. POLYS AND STORE WITH ALL ORDERS
C FOR ONE ANGLE ON RECORD
C
  IP=8
  IP2=9
  REWIND IP
  REWIND IP2
  DO 41 J=1,181
  P(1)=1.0
  P(2)=X(J)
  DO 40 L=3,LESS
  FL=L
  P(L)=((2.0*FL-3.0)*(X(J)*P(L-1))-(FL-2.0)
1 *P(L-2))/(FL-1.0)
40 CONTINUE
  WRITE(IP)(P(L),L=1,LESS)
41 CONTINUE
  REWIND IP
C
C PRODUCE LEG. POLYS. AND STORE WITH ALL ANGLES FOR
C ONE ORDER ON RECORD
C
  IP3=10
  REWIND IP3
  DO 135 J=1,181
135 P(J)=1.0
  WRITE(IP3)(P(J),J=1,181)
  DO 136 J=1,181
  P2(J) = P(J)
136 P(J) = X(J)
  WRITE(IP3)(P(J),J=1,181)
  DO 141 L=3,LESS
  FL = L
  DO 140 J=1,181
  P3(J)=P2(J)
  P2(J)=P(J)
  P(J)=((2.*FL-3.)*(X(J)*P2(J))-(FL-2.)*P3(J))/(FL-1.)
140 CONTINUE
  WRITE(IP3)(P(J),J=1,181)

```

```

141 CONTINUE
    REWIND IP3
    GO TO 46
42 IF(LESS.LE.LESWAS) GO TO 46
    L1=LESWAS+1
    DO 45 J=1,181
    READ(IP)(P(L),L=1,LESWAS)
    DO 44 L=L1,LESS
    FL=L
    P(L)=((2.0*FL-3.0)*(X(J)*P(L-1))-(FL-2.0)
1      *P(L-2))/(FL-1.0)
44 CONTINUE
    WRITE(IP2)(P(L),L=1,LESS)
45 CONTINUE
    ITEMP=IP
    IP=IP2
    IP2=ITEMP
    REWIND IP
    REWIND IP2
    LWM=LESWAS-2
    DO 144 L=1,LWM
144 READ(IP3)
    READ(IP3)(P2(J),J=1,181)
    READ(IP3)(P(J),J=1,181)
    DO 148 L=L1,LESS
    FL=L
    DO 146 J=1,181
    P3(J)=P2(J)
    P2(J)=P(J)
    P(J)=((2.*FL-3.)*(X(J)*P2(J))-(FL-2.)*P3(J))/(FL-1.)
146 CONTINUE
    WRITE(IP3)(P(J),J=1,181)
148 CONTINUE
    REWIND IP3
46 LESWAS = LESS
    CALL CALCF(AN,LESS,F)
    IF(I.GT.0) RETURN
C
C EVALUATE F USING TRUNCATED SERIES
C
C IF(I4.EQ.1)CALL LEGCHK(AN,LESS)
C
C USE SMOOTHED VALUES OF F TO GET LOWER ORDER FIT
C
    IF(I2.EQ.0)GO TO 171
    CALL SMOOTH(F,N,TW,FFIT,NFIT)
    L=1
163 SUM= 0.0
    READ(IP3)(P(J),J=1,181)
    DO 165 J2=1,NFIT
    NPT=N(J2)
    XJ=X(NPT)
    SINJ=SQRT(1.-XJ*XJ)
    SUM=SUM+SINJ*FFIT(J2)*TW(J2)*P(NPT)
165 CONTINUE
    ELM=L-1
    SUM=SUM*(2.*ELM+1.)*2.*PI
    AN2(I)=SUM

```

```

L=L+1
IF(L.LE.LESS) GO TO 163
REWIND IP3
WRITE(6,170)(AN2(L),L=1,LESS)
170 FORMAT(/,'COEFS FROM SMOOTHED F',/,(1P6E15.6))
CALL CALCF(AN2,LESS,F?)
IF(I4.EQ.1)CALL LEGCHK(AN2,LESS)
C
C
C INTEGRATE F OVER ALL ANGLES. RECALCULATE AN(L) VALUES
C (F NOT SMOOTHED)
C
1/1 IF(I3.EQ.0)RETURN
L=1
WRITE(6,62)
62 FORMAT(/,'RECALCULATED COEFFICIENTS - OPTION AGIN.',/,'
1 '(SIMILAR TO 1 DEG SMOOTHING.)')
63 SUM=0.0
READ(IP3)(P(J),J=1,181)
DO 65 J=1,180
COSJ=(X(J)+X(J+1))/2.
SINJ=SQRT(1.-COSJ*COSJ)
FJ=(F(J)+F(J+1))/2.
ELM = L-1
PWT=(2.*ELM+1.)*(P(J)+P(J+1))/2.
SUM=SUM+SINJ*FJ*T1*PWT
65 CONTINUE
SUM= SUM*2.*PI
WRITE(6,70)L,SUM,AN(L)
70 FORMAT('L,SUM,AN(L) = ',I5,1P2E15.6)
AN3(L)=SUM
IF(I,NE.0) RETURN
L=L+1
IF(L.LE.LESS) GO TO 63
REWIND IP3
CALL CALCF(AN3,LESS)
C
C TRY TRUNCATING THE RECALCULATED SERIES
C
IF(I4.EQ.1)CALL LEGCHK(AN3,LESS)
RETURN
END
SUBROUTINE LEGCHK(AN,LESS)
COMMON ALPHA,PI,IP,IP2,IP3
DIMENSION FLO(181),LORDER(7),AN(500),LTEST(6),P(500)
C
C CHECK FOR LOWER LEGENDRE ORDER ACCURACY
C
WRITE(6,2)LESS
2 FORMAT(/,'TRUNCATE TO LESS THAN',I4,' TERMS. (OPTION TRUN)',/,'
1 'TEST FOR')
LORDER(1)=0
C
CCCCCCCCCCCCCCCC
C
C FIND TRUNCATION TEST POINTS
C

```

```

DO 5 I=2,6
5 LTEST(I)=1
  ITEST=1
  LTEST(1)=2.*ALPHA+3.
  WRITE(6,8)LTEST(1)
8 FORMAT(15,' (2*ALPHA+2)')
C
C LOOK FOR LOCAL MINIMUM
C
  L2=LESS/2
6 DO 10 L=L2,LESS
  IF(AN(L).GT.AN(L-1).AND.AN(L).GT.AN(L-2)) GO TO 12
10 CONTINUE
  GO TO 15
12 ITEST=ITEST+1
  LTEST(ITEST)=L
  WRITE(6,13)L
13 FORMAT(15,' (LOCAL MINIMUM)')
  IF(ITEST.EQ.5) GO TO 15
  L2=L+1
  GO TO 6
C
C LOOK FOR FIRST COEFFICIENT.LT.AN(1)
C
15 DO 20 L=2,LESS
  IF(AN(L).LT.AN(1)) GO TO 22
20 CONTINUE
22 ITEST=ITEST+1
  LTEST(ITEST)=L
  WRITE(6,23)L
23 FORMAT(15,' (LT.FIRST COEF)')
C
C PUT TEST POINTS IN ORDER
C
  IF(ITEST.EQ.1) GO TO 27
  ILOOP=0
  ITESTM=ITEST-1
24 IF(ILOOP.GT.15) STOP
  DO 26 I=1,ITESTM
  IF(LTEST(I).LE.LTEST(I+1)) GO TO 26
  ITEMP=LTEST(I)
  LTEST(I)=LTEST(I+1)
  LTEST(I+1)=ITEMP
  ILOOP=ILOOP+1
  GO TO 24
26 CONTINUE
C
C CALCULATE RESULTING PHASE FUNCTIONS
C
27 DO 28 I=1,ITEST
  LORDER (I+1)=LTEST(I)
28 CONTINUE
C
  DO 125 J=1,181
125 FLO(J)=0.0
C
  DO 140 LO=1,ITEST
  L1=LORDER(LO)+1

```

```

L2=LORDER(L0+1)
DO 155 J=1,181
READ(IP)(P(L),L=1,LESS)
DO 150 L=L1,L2
FLO(J)=FLO(J)+AN(L)*P(L)/(4.*PI)
150 CONTINUE
155 CONTINUE
REWIND IP
C
WRITE(6,156)L2,(FLO(J),J=1,181)
156 FORMAT(/,'L = ',I4,/,1PE15.6,/, (1P6E15.6))
140 CONTINUE
C
RETURN
END
SUBROUTINE CALCF(AN,LESS,F)
COMMON XYZ,PI,IP,IP2,IP3
DIMENSION P(500),F(181),AN(500)
DO 55 J=1,181
F(J)=0.0
READ(IP)(P(L),L=1,LESS)
DO 50 L=1,LESS
F(J)=F(J)+AN(L)*P(L)
50 CONTINUE
F(J)=F(J)/(4.*PI)
55 CONTINUE
REWIND IP
WRITE(6,60)(F(J),J=1,181)
60 FORMAT(/,'F FROM LEG. POLYS., 0 TO 180 DEGREES',/,1PE15.6,
1/, (1P6E15.6))
RETURN
END
SUBROUTINE SMOOTH(F,N,TW,FFIT,NFIT)
DIMENSION F(181),N(181),TW(181),FFIT(181)
T1=3.1415926536/180.
WRITE(6,245)
245 FORMAT(/,'SMOOTHING PARAMETERS')
READ(5,246)THRESHF,WIDTH
246 FORMAT(2E10.0)
WRITE(6,247)THRESHF,WIDTH
247 FORMAT(/,'THRESHF,WIDTH=',1P2E15.6)
DO 248 J=2,90
F1=F(1)*THRESHF
IF(F(J).LT.F1) GO TO 249
248 CONTINUE
WRITE(6,250)
250 FORMAT(/,'NO VALUE OF PHASE FUNCT IN FIRST 90 DEG LESS THAN',/,
1 'F(1)*THRESHF TO BEGIN SMOOTHING')
GO TO 61
249 JWIDE=WIDTH
WIDTH=JWIDE
NINT=180./WIDTH
DO 251 I=1,NINT
JLAST=I*JWIDE+1
IF(J.LT.JLAST) GO TO 255
251 CONTINUE
WRITE(6,252) J
252 FORMAT(/,'THRESHOLD ANGLE',I4,' NOT FOUND')
GO TO 61

```

```

255 NFIT=J+1+NINT-I
WRITE(6,253)
253 FORMAT(/,'REP. PT. INDEX WIDTH F')
N(1)=1
TW(1)=T1/2.
FFIT(1)=F(1)
DO 256 K=2,J
N(K)=K
TW(K)=T1
256 FFIT(K)=F(K)
WRITE(6,254)(N(K),TW(K),FFIT(K),K=1,J)
254 FORMAT(15,2E15.6)
N(J+1)=(JLAST+J)/2
FFITJ=0.0
JM=JLAST-1
IF(N(J+1).EQ.J)N(J+1)=J+1
IF(J.EQ.JM)GO TO 357
DO 257 J2=J+1,JM
257 FFITJ=FFITJ+F(J2)
357 FFITJ=FFITJ+F(JLAST)/2.
POINTS=FLOAT(JLAST-J)-0.5
FFIT(J+1)=FFITJ/POINTS
TW(J+1)=POINTS*T1
WRITE(6,254)N(J+1),TW(J+1),FFIT(J+1)
JMID=JLAST-JWIDE/2
DO 260 K=J+2,NFIT
JMID=JMID+JWIDE
N(K)=JMID
JFIRST=JLAST
JLAST=JFIRST+JWIDE
FFITK=(F(JFIRST)+F(JLAST))/2.
IF(JWIDE.EQ.1) GO TO 259
JP=JFIRST+1
JM=JLAST-1
DO 258 J=JP,JM
258 CONTINUE
259 FFIT(K)=FFITK/FLOAT(JWIDE)
TW(K)=T1*FLOAT(JWIDE)
WRITE(6,254)N(K),TW(K),FFIT(K)
260 CONTINUE
61 RETURN
END

```

APPENDIX B. LEGEDIT UTILIZATION INSTRUCTIONS

The LEGEDIT code lists, orders, and deletes sets of microscopic Legendre coefficients on the LEG file. It can be included in a run after MIELEG to order the file produced from a pre-existing file and the new MIELEG contributions, or it can be run by itself.

B.1 LEGEDIT Input Description

MODE (15)

LEGEDIT input data consists of a single integer variable, MODE. The options are:

MODE = 0 Identify entries on files LEG, NEWLEG, BACKLEG, OTHERLEG

LEG - Normal Legendre coefficient file

NEWLEG - Temporary production LEG file during MIELEG run.

BACKLEG - Back-up LEG file, produced at beginning of MIELEG run.

OTHERLEG - Overflow file for other refractive indices, β . Contains entries edited from LEG file which did not have same refractive index as originally found on file.

MODE = 1 Ordering mode. Puts entries in order of ascending size parameter, α . Deletes old entry if value of α within 0.01 of new entry. Writes on OTHERLEG any entries with refractive index, β , different from that originally on file.

MODE = 2 List mode. List LEG file

MODE = 12 Order, then list.

MODE = 21 List, then order

MODE = 212 List, order, then list again.

MODE = 3 Deletes selected entries.

ALFOUT (E10.0) Ignored unless MODE = 3

Up to 10 selected deletions, input one per record. Entries will be deleted if size parameter, α , within 0.01 of ALFOUT.

B.2 LEGEDIT Output Description - Printout

Identification mode 0 prints size parameter (α , ALPHA), complex refractive index (β , PM) cross-section ratio (K, SCKRO), and order (N, LESS) for each entry of LEG, NEWLEG, BACKLEG, OTHERLEG. Ordering mode 1 (or the ordering part of 12, 21, or 212) lists the size parameter, complex refractive index, cross-section ratio, and Legendre order of each entry on file LEG. This list is produced once before the size parameter ordering and deletions occur, and again after they occur.

List mode 2 (or the listing part of 12, 21, or 212) lists the size parameter, refractive index, cross-section ratio, and Legendre order of each entry on LEG, followed by the set of coefficients associated with these parameters. Mode 3 (selected deletions) operation is followed by a list of the amended file.

B.3 LEGEDIT Output Description - Files

The purpose of the ordering mode of LEGEDIT is to produce a LEG file in ascending order of the size parameter. In addition, the LEGEDIT job (OR MIELEG followed by LEGEDIT) saves a backup file, BACKLEG, which is a copy of the original LEG file. The updated LEG file is produced on a file named NEWLEG, so an incomplete job might have the required data on NEWLEG. A successfully completed LEGEDIT job would produce identical LEG and NEWLEG files.

B.4 LEGEDIT FORTRAN Listing


```

C
C LEGEDIT -- CODE TO EDIT LIBRARY OF LEGENDRE COEFFICIENTS FROM MIELEG
C
C
C   READ(5,10) MODE
C   10 FORMAT(I5)
C
C MODE DETERMINES EDITING TYPE
C   =0 TO READ FILES 11,12,13,14 (LEG,NEWLEG,BACKLEG,OTHERLEG)
C   =1 TO ORDER AND DELETE DUPLICATES (ALWAYS DELETE WITH ORDER),
C   WRITE ENTRIES WITH DIFFERENT REFR. IND. ON OTHERLEG
C   =2 TO LIST ENTIRE FILE
C   =12 TO ORDER THEN LIST
C   =21 TO LIST THEN ORDER
C   =212 TO LIST, ORDER, THEN LIST AGAIN
C   = 3 TO DELETE SELECTED ENTRIES
C
C   REWIND 11
C   REWIND 12
C   IF(MODE.EQ.3)CALL SCRUB
C   IF(MODE.EQ.0) CALL READIT
C   IF(MODE.EQ.1) CALL ORDER
C   IF(MODE.EQ.2) CALL LIST(11,12)
C   IF(MODE.NE.12) GO TO 21
C   CALL ORDER
C   CALL LIST(12,12)
C 21 IF(MODE.NE.21) GO TO 212
C   CALL LIST(11,11)
C   CALL ORDER
C 212 IF(MODE.NE.212) STOP
C   CALL LIST(11,11)
C   CALL ORDER
C   CALL LIST(12,12)
C   STOP
C   END
C   SUBROUTINE READIT
C   FILE 11 (LEG) - NORMAL LIBRARY (SINGLE IND. OF REFR.)
C   FILE 12 (NEWLEG) - NEW ADDITIONS TO LIBRARY, MAY NOT BE
C   OPEN (IF LAST ADDITION EXITED NORMALLY)
C   FILE 13 (BACKLEG) - LEG BACKUP
C   FILE 14 (OTHERLEG) - RECEPTACLE FOR DATA WITH OTHER INDICES OF REFR.
C   COMPLEX PM
C   DO 50 I=1,4
C   IN = 10 + I
C   WRITE(6,19)IN
C 19 FORMAT(/,'CONTENTS OF FILE',I3)
C 20 READ(IN,END=50)ALPHA,PM,SCKRO,LESS
C   WRITE(6,15)ALPHA,PM,SCKRO,LESS
C 15 FORMAT(/,F7.2,5X,2F7.2,5X,1PE15.6,I10)
C   READ(IN)
C   GO TO 20
C 50 CONTINUE
C   REWIND IN
C   RETURN
C   END
C   SUBROUTINE ORDER
C   COMPLEX PM(200),P,PLAST
C   DIMENSION ALPHA(200),LESS(200),IORDER(200),COEF(500),ITAG(200)

```

```

1 .SCKRO(200)
C
  REWIND 14
  I=0
  IOTHER = 0
  INEWAS=0
10 I = I + 1
  READ(11,END=50) ALPHA(I),PM(I),SCKRO(I),LESS(I)
  READ(11)
  IF(I.LT.200)GO TO 10
  WRITE(6,15)
15 FORMAT(/,'NUMBER OF FILES EXCEEDS 200.  DIMENSIONS TOO SMALL.')
  RETURN
C
50 ILAST = I - 1
  PLAST=PM(ILAST)
  WRITE(6,55)(I,ALPHA(I),PM(I),SCKRO(I),LESS(I),I#1,ILAST)
55 FORMAT(/,11X,'ALPHA',4X,'IND. OF REFR.',8X,'X-S RATIO',4X
1  'ORDER ON LIB FILE'/
2  (14,5X,0PF7.2,5X,2F7.2,5X,1PE15.6,I8))
C
  DO 57 I=1,ILAST
  ITAG(I)=1
  IORDER(I)=1
57 CONTINUE
C
  ISAVE = ILAST
  ILASTM=ILAST-1
58 I1 = 1
59 DO 60 I=I1,ILASTM
  IF(PM(I).EQ.PLAST)GO TO 62
  IF(ITAG(I).EQ.0)GO TO 60
  WRITE(14)ALPHA(I),PM(I),SCKRO(I),LESS(I)
  WRITE(14)(COEF(N),N=1,LESS(I))
  WRITE(6,61)I
61 FORMAT(' ENTRY NO.',I4,' WRITTEN TO FILE OTHERLEG SINCE IND. OF'
1, ' REFR DIFFERENT.')
  IOTHER=1
  ITAG(I)=0
  GO TO 63
62 IF(ABS(ALPHA(I)-ALPHA(I+1)).LT.0.01)GO TO 68
  IF(ALPHA(I)-ALPHA(I+1))60,68,65
68 IF(ITAG(I+1).EQ.0)GO TO 60
  ITAG(I+1)=0
C
63 ISAVE=ISAVE-1
60 CONTINUE
  IF(I1.GT.1) GO TO 58
  GO TO 70
65 ITEMP = IORDER(I)
  ITAGT=ITAG(I)
  ATEMP=ALPHA(I)
  IORDER(I)=IORDER(I+1)
  ITAG(I)=ITAG(I+1)
  ALPHA(I)=ALPHA(I+1)
  IORDER(I+1) = ITEMP
  ITAG(I+1)=ITAGT
  ALPHA(I+1)=ATEMP

```

```

C      IF(I.EQ.ILAST)GO TO 58
66     I1 = I + 1
      GO TO 59
C
70     REWIND 11
      REWIND 12
      REWIND 14
C
      INEW = 1
      WRITE(6,71)
71     FORMAT(//,'REORDERED AS FOLLOWS. (DELETIONS SKIPPED.)')
69     DO 80 I=1,ILAST
      IF(IORDER(INEW).NE.I) GO TO 75
      IF(ITAG(INEW).EQ.0) GO TO 73
      READ(11)A,P,SK,L
      READ(11)(COEF(N),N=1,LESS(I))
      WRITE(6,72)INEW,A,P,SK,L
72     FORMAT(I4,5X,F7.2,5X,2F7.2,5X,1PE15.6,18)
      WRITE(12) A,P,SK,L
      WRITE(12)(COEF(N),N=1,LESS(I))
      INEW = INEW + 1
      GO TO 74
73     READ (11)
      READ (11)
      INEW = INEW + 1
74     IF(INEW.GT.ILAST) GO TO 90
      GO TO 80
75     READ(11)
      READ(11)
80     CONTINUE
      REWIND 11
      IF(INEW.GT.ILAST)GO TO 90
      IF(INEW.EQ.INEWAS) GO TO 82
      INEWAS=INEW
      GO TO 69
82     WRITE(6,84)INEW
84     FORMAT(/,'CANNOT FIND NO.',I4,' STOP.')
      RETURN
90     J=0
      IF(IOTHER.EQ.0)GO TO 110
      WRITE(6,92)
92     FORMAT(/,'ENTRIES ON OTHERLEG')
95     READ(14,END=110)A,P,SK,L
      J=J+1
      READ(14)(COEF(N),N=1,LESS(I))
      WRITE(6,72)J,A,P,SK,L
      GO TO 95
110    REWIND 11
      REWIND 12
      REWIND 14
      RETURN
      END
      SUBROUTINE LIST(IN,IOUT)
      COMPLEX PM
      DIMENSION COEF(500)
      WRITE(6,10)

```

```

10 FORMAT(/, 'CONTENTS OF LEG LIBRARY FILE')
   I=0
12 READ(IN,END=30)ALPHA,PM,SCKRO,LESS
   READ(IN)(COEF(N),N=1,LESS)
   I=I+1
   IF(IN.EQ.IOUT) GO TO 14
   WRITE(IOUT)ALPHA,PM,SCKRO,LESS
   WRITE(IOUT)(COEF(N),N=1,LESS)
14 WRITE(6,15) I,ALPHA,PM,SCKRO,LESS
15 FORMAT(/, 'ENTRY', I3,5X, 'ALPHA='F7.2,5X, 'PM='F7.2,5X,
   1 'SCKRO='F15.6,5X, 'LESS='I4)
   WRITE(6,20)(COEF(N),N=1,LESS)
20 FORMAT(/, (1P6E15.6))
   GO TO 12
30 REWIND IN
   REWIND IOUT
   RETURN
   END
   SUBROUTINE SCRUB
   DIMENSION ALFOUT(10),COEF(500)
   COMPLEX PM
   DO 10 I=1,10
   READ(5,5,END=15)ALFOUT(I)
10 CONTINUE
   5 FORMAT(E10.0)
15 IOUT=I-1
   WRITE(6,16)(ALFOUT(I),I=1,IOUT)
16 FORMAT(/, 'SCRUB LEG ENTRIES WITH ALPHA:'/(F10.2))
20 READ(11,END=50)ALPHA,PM,SCKRO,LESS
   READ(11)(COEF(N),N=1,LESS)
   DO 25 I=1,IOUT
   DALF=ABS(ALFOUT(I)-ALPHA)
   IF(DALF.LT.0.01) GO TO 20
25 CONTINUE
   WRITE(12)ALPHA,PM,SCKRO,LESS
   WRITE(12)(COEF(N),N=1,LESS)
   GO TO 20
50 REWIND 12
   CALL LIST(12,12)
   STOP
   END

```

APPENDIX C. MACRO UTILIZATION INSTRUCTIONS

The MACRO code produces a set of macroscopic Legendre coefficients from the microscopic sets on the LEG file. MACRO is essentially identical to parts of the MIELEG code, which produces macroscopic coefficients directly from basic parameters of wavelength, refractive index, and particle size distribution and range.

C.1 MACRO Input Description

TITLE (20A4) Any descriptive alphanumeric information.

RMIN, RMAX, PM(Complex), WAVEL, NEQ (5F10.0, I5)

- RMIN - Minimum particle radius considered, microns
- RMAX - Maximum particle radius considered, microns
- PM - Complex refractive index, β , composed of two parts:
 - a) M, real index of refraction relative to the surrounding material, and
 - b) K, extinction coefficient of the particle material.
- WAVEL - Wavelength of the radiation, microns
- NEQ - Number of equation describing particle size distribution, 0 or 1
 - 0: $N(R) = P1*R**P2*EXP(-P3*R**P4)$
 - 1: $N(R) = P1$ for $R \leq P2$
 $N(R) = P3*R**P4$ for $R > P2$

P1, P2, P3, P4 (4F10.0)

Parameters described by choice of NEQ, on previous input record.

Output Options 4(1X, A4)

Possible options:

XSOR or XSLA, TRUN, AGIN, SMOO

XSOR - Produces macroscopic Legendre coefficients appropriately normalized for use in Oak Ridge code (ANISN, TDA, DOT, MORSE).

XSLA - Produced macroscopic Legendre coefficients appropriately normalized for use in Los Alamos code (DTF, TRANZIT, TWOTRAN).

TRUN - Truncation option to test expansion of Legendre series to fewer than total number of coefficients available.

AGIN - Refitting option to reproduce coefficients from expansion, testing consistency of method.

SMOO - Smoothing option to refit from smoothed data.

THRESHF, WIDTH (2E10.0) - Smoothing parameters, used with SMOO option.

See MIELEG Utilization Instructions for a more complete discussion of output options.

C.2 MACRO Output Description - Printout

The MACRO code prints out all input data. It then prints the size parameters of all data from the LEG file appropriate to the wavelength and size range of the problem, along with the particle size, range, and relative density associated with each of the data points. It then prints the complete LEG file entry for each of these data points.

If the TRUN option is used, the expansion is printed for each truncation. If the SMOO and AGIN options are used, the resulting coefficients are printed along with the full expansion. The TRUN option, along with SMOO or AGIN also causes the truncated expansions for these cases to be printed.

C.3 MACRO Output Description - Files

The MACRO code produces a file, MAC, containing Legendre coefficients integrated with particle size distribution. This file can be manipulated by CRT, merging into input data files for transport codes. Since other MACRO or MIELEG runs will write over the MAC file, MAC must be copied to some other file if it is to be saved while other MAC data are being produced.

C.4 MACRO FORTRAN Listing

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

MACRO CODE

```

C
C MACRO - CODE TO PRODUCE MACRO X-S COEF. FROM SIZE DISTRIBUTION AND
C LEGENDRE COEF ON FILE 11 (LEG) AND PUT X-S ON FILE 15 (MAC)
C
      COMPLEX P,PM
      DIMENSION COEF(500),COEFXS(500),RI(200),ENR(200),ALPHA(200),
1 SCKRO(200),SIG(200),LI(200),DELR(200),TITLE(20)
      DATA IA0,IA1,IA2,IA3,IA4/4HXSOR,4HXSLA,4HSMO0,4HAGIN,4HTRUN/
      DATA I1,I2,I3,I4/4*0/
      COMMON/OUT/I1,I2,I3,I4
      COMMON ALPHER,PI
      PI=3.1415926536
      READ(5,2)(TITLE(I),I=1,20)
2  FORMAT(20A4)
      WRITE(6,4)(TITLE(I),I=1,20)
1  FORMAT(/,20A4)
      DO 10 MM=1,500
10 COEFXS(MM)=0.0

C
C DETERMINE DATA REQUIRED
C
      READ(5,15)RMIN,RMAX,PM,WAVEL,NEQ
15  FORMAT(5F10.0,I5)
      WRITE(6,16)RMIN,RMAX,PM,WAVEL
16  FORMAT(/,'RMIN,RMAX='2F7.2,' MICRONS (E-04 CM)',/
1  'INDEX OF REFR='2F7.2,' (COMPLEX)',/
2  'WAVELENGTH ='F7.2)
      WAVEN=2.*PI/WAVEL
      AMIN = WAVEN * RMIN
      AMAX = WAVEN * RMAX
      WRITE(6,20) AMIN,AMAX
20  FORMAT(/,'ALPHA RANGE ='1,2F7.2)

C
C SEARCH LEG FILE FOR DATA
C
      REWIND 11
      REWIND 12
      LESS=0
      I=0
25  READ(11,END=50) A,P,SK,L
      IF(A.GT.AMAX) GO TO 50
      IF(A.LT.AMIN) GO TO 35
      IF(P.NE.PM) GO TO 35
      I = I + 1
      ALPHA(I) = A
      XYZ=A
      ALPHER=A
      SCKRO(I)=SK
      LI(I)=L
      IF(L.GT.LESS) LESS=L
      READ(11) (COEF(MM),MM=1,L)
      WRITE(12)(COEF(MM),MM=1,L)
      GO TO 25
35  READ(11)
      GO TO 25
50  NRS = I
      MORDER=2.*ALPHER+3.
      REWIND 11

```

```

REWIND 12
C
C DETERMINE PARTICLE DISTRIBUTION AT DATA POINTS
C
DO 60 I=1,NRS
RI(I)=ALPHA(I)/WAVEN
60 CONTINUE
IF(NRS.GT.1)GO TO 62
DELR(1)=RMAX-RMIN
GO TO 72
62 DO 70 I=1,NRS
IF(I.GT.1.AND.I.LT.NRS) GO TO 110
IF(I.EQ.NRS)DELR(I)=RMAX-(RI(I-1)+RI(I))/2.
IF(I.EQ.1)DELR(I)=(RI(I+1)+RI(I))/2.-RMIN
GO TO 70
110 DELR(I)=(RI(I+1)-RI(I-1))/2.0
70 CONTINUE
72 WRITE(6,112)
112 FORMAT(/,'PARTICLE DISTRIBUTION PARAMETERS')
130 IF(NEQ)120,132,140
132 READ(5,620)P1,P2,P3,P4
WRITE(6,630)P1,P2,P3,P4
DO 137 I=1,NRS
137 ENR(I)=P1*RI(I)**P2*EXP(-P3*RI(I)**P4)
GO TO 150
140 READ(5,620)P1,P2,P3,P4
WRITE(6,640)P1,P2,P3,P4
DO 118 I=1,NRS
IF(RI(I).GT.P2) GO TO 157
ENR(I)=P1
GO TO 118
157 ENR(I)=P3*RI(I)**P4
118 CONTINUE
GO TO 150
120 WRITE(6,125)
125 FORMAT(/,'DO NOT KNOW WHAT TO DO WITH NEGATIVE NEQ. STOP.')
STOP
620 FORMAT(4F10.0)
630 FORMAT(/,2X,'P1 = ',1PE15.6,/,2X,'P2 = ',1PE15.6,/,2X,
1 'P3 = ',1PE15.6,/,2X,'P4 = ',1PE15.6)
640 FORMAT(/,2X,'P1 = ',1PE15.6,/,2X,'P2 = ',1PE15.6,/,2X,
1 'P3 = ',1PE15.6,/,2X,'P4 = ',1PE15.6)
150 WRITE(6,200)
200 FORMAT(/,'DATA POINTS'/,5X,'ALPHA',8X,'R',8X,'DELR',13X,'ENR')
WRITE(6,220)(I,ALPHA(I),RI(I),DELR(I),ENR(I),I=1,NRS)
220 FORMAT(I3,F7.2,F10.2,2E15.6)
C
C CALCULATE MEAN MACRO X-S VALUES
C
EN=0.0
DO 250 I=1,NRS
SIG(I)=SCKRO(I)*ENR(I)*PI*RI(I)*RI(I)*DELR(I)*1.0E-08
EN=EN+ENR(I)*DELR(I)
L=LI(I)
READ(12)(COEF(MM),MM=1,L)
WRITE(6,225)I,ALPHA(I),SCKRO(I),SIG(I)

```

```

225 FORMAT(/,'DATA FOR NO.',I3,/, 'ALPHA=',F7.2,3X,'SCKRO=',
1 E15.6,3X,'SIG=',E15.6)
WRITE(6,230)(COEF(MM),MM=1,L)
230 FORMAT(/,'COEFFICIENTS',/, (1P6E15.6))
DO 240 MM=1,LESS
COEFXS(MM)=COEFXS(MM)+COEF(MM)*SIG(I)
240 CONTINUE
250 CONTINUE
C
C WRITE RESULTS
C
WRITE(6,290)EN
290 FORMAT(/,'MACRO X-S COEFFICIENTS',/, 'PARTICLE DENSITY = 'E15.6)
WRITE(6,300)(COEFXS(MM),MM=1,LESS)
300 FORMAT(/, (1P6E15.6))
C
C OTHER OUTPUT
C
READ(5,302)IOUT1,IOUT2,IOUT3,IOUT4
302 FORMAT(4(1X,A4))
C
C CHECK HOLLERITH VALUES
C
WRITE(6,303)IOUT1,IOUT2,IOUT3,IOUT4,IA0,IA1,IA2,IA3,IA4
303 FORMAT(/,'OPTIONS CHOSEN: ',4(1X,A4),/, 'AVAILABLE OPTIONS:',
1 5(1X,A4))
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C I1=1 IF X-S SET FOR ANISN (OR OTHER OAK RIDGE CODE) PRODUCED
IF(IOUT1.EQ. IA0.OR. IOUT2.EQ. IA0.OR. IOUT3.EQ. IA0.
1 OR. IOUT4.EQ. IA0) I1=1
I1WAS=I1
C I1=2 IF X-S SET FOR DTF (OR OTHER LOS ALAMOS CODE) PRODUCED
IF(IOUT1.EQ. IA1.OR. IOUT2.EQ. IA1.OR. IOUT3.EQ. IA1.
1 OR. IOUT4.EQ. IA1) I1=2
C I2=1 IF SMOOTHING TESTED
IF(IOUT1.EQ. IA2.OR. IOUT2.EQ. IA2.OR. IOUT3.EQ. IA2.
1 OR. IOUT4.EQ. IA2) I2=1
C I3=1 IF REFITTING OF COEFS TO F TESTED
IF(IOUT1.EQ. IA3.OR. IOUT2.EQ. IA3.OR. IOUT3.EQ. IA3.
1 OR. IOUT4.EQ. IA3) I3=1
C I4=1 IF TRUNCATION TESTED
IF(IOUT1.EQ. IA4.OR. IOUT2.EQ. IA4.OR. IOUT3.EQ. IA4.
1 OR. IOUT4.EQ. IA4) I4=1
C
IF(I1.EQ.0) GO TO 288
IF(I1WAS.EQ.1.AND. I1.EQ.2) GO TO 270
GO TO 288
270 I1=1
WRITE(6,272)
272 FORMAT(/,'CANNOT PRODUCE O.R. AND L.A. FORMAT X-S '
1 'SETS IN THE SAME RUN. O.R. (E.G., ANISN) CHOSEN. ')
287 FORMAT(' OUTPUT INDICES:',I5,5X ,3I5)
289 FORMAT(' OUTPUT INDICES: ',5X,4I5)
288 IF(I1.EQ.0) GO TO 305
IF(I1.EQ.1)GO TO 285
DO 280 MM=1,LESS
COEFXS(MM)=COEFXS(MM)/(2.* FLOAT(MM)+1.)

```

```

280 CONTINUE
    WRITE(6,289) I1,I2,I3,I4
    GO TO 281
285 WRITE(6,287) I1,I2,I3,I4
281 ZERO=0.0
    WRITE(15,4)(TITLE(I),I=1,20)
    WRITE(15,291) RMIN,RMAX,PM,WAVEL,NEQ,P1,P2,P3,P4,NRS,LESS,EN,MORDER
291 FORMAT(/,'PARAMETERS',/,5E15.6,/,I4,4E15.6,/,2I5.E15.6,I5)
    WRITE(15,292) ZERO,ZERO,COEFS(1),COEFS(1),
      1 (ZERO,ZERO,ZERO,COEFS(MM),MM=2,LESS)
292 FORMAT(/,'6E12.5 FORMAT CROSS-SECTIONS',/, '14**'./2F12.1,
      1 2E12.5,/, (3F12.1,E12.5))
    IF(I1.EQ.1) WRITE(6,293)
293 FORMAT(/,'MACR X-S DATA WRITTEN ON MAC FILE (15). - OPTION XSOR')
    IF(I1.EQ.2) WRITE(6,294)
294 FORMAT(/,'MACR X-S DATA WRITTEN ON MAC FILE(15). - OPTION XSLA')
C
305 CALL TESTAN(0,COEFS,LESS)
    STOP
    END
    SUBROUTINE TESTAN(I,AN,LESS)
    DIMENSION X(181),AN(500),AN2(500),AN3(500),P(500),
      1 P2(500),P3(181),F(181),F2(181),F3(181),N(181),TW(181),
      2 FFIT(181)
    COMMON ALPHER,PI,IP,IP2,IP3
    COMMON/OUT/I1,I2,I3,I4
C
    IF(LESS.LT.501) GO TO 10
    WRITE(6,5) LESS
      5 FORMAT(/,'LESS = ',I4,' MUST CHANGE DIMENSION IN TESTAN AND LEGCHK
      1')
    STOP
10 IF(I.GT.1) GO TO 42
    T1=PI/180.
    DO 30 J=1,181
      T=T1*(J-1)
      X(J)=COS(T)
30 CONTINUE
C
C PRODUCE LEG. POLYS AND STORE WITH ALL ORDERS
C FOR ONE ANGLE ON RECORD
C
    IP=8
    IP2=9
    REWIND IP
    REWIND IP2
    DO 41 J=1,181
      P(1)=1.0
      P(2)=X(J)
      DO 40 L=3,LESS
        FL=L
        P(L)=((2.0*FL-3.0)*(X(J)*P(L-1))-(FL-2.0)
          1 *P(L-2))/(FL-1.0)
C
C RECURSION WITH SMALLER ROUND-OFF ERROR TESTED:
C XJ=X(J)
C PL1=P(L-1)
C PL2=P(L-2)
C P(L)=XJ*PL1-PL2+XJ*PL1-(XJ*PL1-PL2)/(FL-1.0)

```

```

C
C DID NOT APPRECIABLY REDUCE REFIT COEFFICIENTS PAST
C 2*ALPHA+2, THEREFORE NOT DUE TO ROUND-OFF
40 CONTINUE
  WRITE(IP)(P(L),L=1,LESS)
41 CONTINUE
  REWIND IP
C
C PRODUCE LEG. POLYS. AND STORE WITH ALL ANGLES FOR
C ONE ORDER ON RECORD
C
  IP3=10
  REWIND IP3
  DO 135 J=1,181
135 P(J)=1.0
  WRITE(IP3)(P(J),J=1,181)
  DO 136 J=1,181
  P2(J) = P(J)
136 P(J) = X(J)
  WRITE(IP3)(P(J),J=1,181)
  DO 141 L=3,LESS
  FL = L
  DO 140 J=1,181
  P3(J)=P2(J)
  P2(J)=P(J)
  P(J)=((2.*FL-3.)*(X(J)*P2(J))-(FL-2.)*P3(J))/(FL-1.)
C
C SECOND VERSION OF ALTERNATIVE RECURSION:
C
C XJ=X(J)
C P2J=P2(J)
C P3J=P3(J)
C P(J)=XJ*P2J-P3J+XJ*P2J-(XJ*P2J-P3J)/(FL-1.0)
C
C EXPECT RESIDUALLY LARGE VALUES OF HIGH-ORDER REFITTED
C COEFFICIENTS DUE TO INSUFFICIENT INFORMATION IN FUNCTION
140 CONTINUE
  WRITE(IP3)(P(J),J=1,181)
141 CONTINUE
  REWIND IP3
  GO TO 46
42 IF(LESS.LE.LESWAS) GO TO 46
  L1=LESWAS+1
  DO 45 J=1,181
  READ(IP)(P(L),L=1,LESWAS)
  DO 44 L=L1,LESS
  FL=L
  P(L)=((2.0*FL-3.0)*(X(J)*P(L-1))-(FL-2.0)
1 *P(L-2))/(FL-1.0)
44 CONTINUE
  WRITE(IP2)(P(L),L=1,LESS)
45 CONTINUE
  ITEMP=IP
  IP=IP2
  IP2=ITEMP
  REWIND IP
  REWIND IP2
  LWM=LESWAS-2

```

```

DO 144 L=1,LWM
144 READ(IP3)
   READ(IP3)(P2(J),J=1,181)
   READ(IP3)(P(J),J=1,181)
   DO 148 L=L1,LESS
     FL=L
     DO 146 J=1,181
       P3(J)=P2(J)
       P2(J)=P(J)
       P(J)=((2.*FL-3.)*(X(J)*P2(J))-(FL-2.)*P3(J))/(FL-1.)
146 CONTINUE
   WRITE(IP3)(P(J),J=1,181)
148 CONTINUE
   REWIND IP3
46 LESWAS = LESS
   CALL CALCF(AN,LESS,F)
   IF(I.GT.0) RETURN

C
C EVALUATE F USING TRUNCATED SERIES
C
C   IF(I4.EQ.1)CALL LEGCHK(AN,LESS)
C
C USE SMOOTHED VALUES OF F TO GET LOWER ORDER FIT
C
   IF(I2.EQ.0)GO TO 171
   CALL SMOOTH(F,N,TW,FFIT,NFIT)
   L=1
163 SUM= 0.0
   READ(IP3)(P(J),J=1,181)
   DO 165 J2=1,NFIT
     NPT=N(J2)
     XJ=X(NPT)
     SINJ=SQRT(1.-XJ*XJ)
     SUM=SUM+SINJ*FFIT(J2)*TW(J2)*P(NPT)
165 CONTINUE
   ELM=L-1
   SUM=SUM*(2.*ELM+1.)*2.*PI
   AN2(L)=SUM
   L=L+1
   IF(L.LE.LESS) GO TO 163
   REWIND IP3
   WRITE(6,170)(AN2(L),L=1,LESS)
170 FORMAT(/,'COEFS FROM SMOOTHED F',/, '(1P6E15.6)')
   CALL CALCF(AN2,LESS,F2)
   IF(I4.EQ.1)CALL LEGCHK(AN2,LESS)

C
C
C INTEGRATE F OVER ALL ANGLES, RECALCULATE AN(L) VALUES
C   (F NOT SMOOTHED)
C
171 IF(I3.EQ.0)RETURN
   L=1
   WRITE(6,62)
62 FORMAT(/,'RECALCULATED COEFFICIENTS - OPTION AGIN.',/,
1 '(SIMILAR TO 1 DEG SMOOTHING.))')
63 SUM=0.0
   READ(IP3)(P(J),J=1,181)
   DO 65 J=1,180

```

```

COSJ=(X(J)+X(J+1))/2.
SINJ=SQRT(1.-COSJ*COSJ)
FJ=(F(J)+F(J+1))/2.
ELM = L-1
PWT=(2.*ELM+1.)*(P(J)+P(J+1))/2.
SUM=SUM+SINJ*FJ*T1*PWT
65 CONTINUE
SUM= SUM*2.*PI
WRITE(6,70)L,SUM,AN(L)
70 FORMAT('L,SUM,AN(L) = ',I5,1P2E15.6)
AN3(L)=SUM
IF(I.NE.0) RETURN
L=L+1
IF(L.LE.LESS) GO TO 63
REWIND IP3
CALL CALCF(AN3,LESS)
C
C TRY TRUNCATING THE RECALCULATED SERIES
C
IF(I4.EQ.1)CALL LEGCHK(AN3,LESS)
RETURN
END
SUBROUTINE LEGCHK(AN,LESS)
COMMON ALPHER,PI,IP,IP2,IP3
DIMENSION FLO(181),LORDER(7),AN(500),LTEST(6),P(500)
C
C CHECK FOR LOWER LEGENDRE ORDER ACCURACY
C
WRITE(6,2)LESS
2 FORMAT(/,'TRUNCATE TO LESS THAN',I4,' TERMS. (OPTION TRUN)',/,
1 'TEST FOR')
LORDER(1)=0
C
C CCCCCCCCCCCCCCCC
C
C FIND TRUNCATION TEST POINTS
C
DO 5 I=2,6
5 LTEST(I)=1
ITEST=1
ITEST(1)=2.*ALPHER+3.
WRITE(6,8)LTEST(1)
8 FORMAT(I5,' (2*ALPHA+2)')
C
C LOOK FOR LOCAL MINIMUM
C
L2=LESS/2
6 DO 10 L=L2,LESS
IF(AN(L).GT.AN(L-1).AND.AN(L).GT.AN(L-2)) GO TO 12
10 CONTINUE
GO TO 15
12 ITEST=ITEST+1
LTEST(ITEST)=L
WRITE(6,13)L
13 FORMAT(I5,' (LOCAL MINIMUM)')
IF(ITEST.EQ.5) GO TO 15
L2=L+1
GO TO 6

```

```

C
C LOOK FOR FIRST COEFFICIENT.LT.AN(1)
C
15 DO 20 I=2,LESS
   IF(AN(I).LT.AN(1)) GO TO 22
20 CONTINUE
22 ITEST=ITEST+1
   LTEST(ITEST)=L
   WRITE(6,23)L
23 FORMAT(I5,' (LT.FIRST COEF)')
C
C PUT TEST POINTS IN ORDER
C
   IF(ITEST.EQ.1) GO TO 27
   ILOOP=0
   ITESTM=ITEST-1
24 IF(ILOOP.GT.15) STOP
   DO 26 I=1,ITESTM
   IF(LTEST(I).LE.(LTEST(I+1))) GO TO 26
   ITEMP=LTEST(I)
   LTEST(I)=LTEST(I+1)
   LTEST(I+1)=ITEMP
   ILOOP=ILOOP+1
   GO TO 24
26 CONTINUE
C
C CALCULATE RESULTING PHASE FUNCTIONS
C
27 DO 28 I=1,ITEST
   LORDER (I+1)=LTEST(I)
28 CONTINUE
C
   DO 125 J=1,181
125 FLO(J)=0.0
C
   DO 140 LO=1,ITEST
   L1=LORDER(LO)+1
   L2=LORDER(LO+1)
   DO 155 J=1,181
   READ(IP)(P(L),L=1,LESS)
   DO 150 L=L1,L2
   FLO(J)=FLO(J)+AN(L)*P(L)/(4.*PI)
150 CONTINUE
155 CONTINUE
   REWIND IP
C
   WRITE(6,156)L2,(FLO(J),J=1,181)
156 FORMAT(/,'L = ',I4,/,1PE15.6,/, (1P6E15.6))
140 CONTINUE
C
   RETURN
   END
   SUBROUTINE CALCF(AN,LESS,F)
   COMMON ALPHER,PI,IP,IP2,IP3
   DIMENSION P(500),F(181),AN(500)
   DO 55 J=1,181
   F(J)=0.0
   READ(IP)(P(L),L=1,LESS)

```

```

DO 50 L=1,LESS
F(J)=F(J)+AN(L)*P(L)
50 CONTINUE
F(J)=F(J)/(4.*PI)
55 CONTINUE
REWIND IP
WRITE(6,60)(F(J),J=1,181)
60 FORMAT(/,'F FROM LEG. POLYS., 0 TO 180 DEGREES',/,1PE15.6,
1/.(1P6E15.6))
RETURN
END
SUBROUTINE SMOOTH(F,N,TW,FFIT,NFIT)
DIMENSION F(181),N(181),TW(181),FFIT(181)
T1=3.1415926536/180.
WRITE(6,245)
245 FORMAT(/,'SMOOTHING PARAMETERS')
READ(5,246)THRESHF,WIDTH
246 FORMAT(2E10.0)
WRITE(6,247)THRESHF,WIDTH
247 FORMAT(/,'THRESHF,WIDTH=',1P2E15.6)
DO 248 J=2,90
F1=F(1)*THRESHF
IF(F(J).LT.F1) GO TO 249
248 CONTINUE
WRITE(6,250)
250 FORMAT(/,'NO VALUE OF PHASE FUNCT IN FIRST 90 DEG LESS THAN',/,
1 'F(1)*THRESHF TO BEGIN SMOOTHING')
GO TO 61
249 JWIDE=WIDTH
WIDTH=JWIDE
NINT=180./WIDTH
DO 251 I=1,NINT
JLAST=I*JWIDE+1
IF(J.LT.JLAST) GO TO 255
251 CONTINUE
WRITE(6,252) J
252 FORMAT(/,'THRESHOLD ANGLE',I4,' NOT FOUND')
GO TO 61
255 NFIT=J+1+NINT-I
WRITE(6,253)
253 FORMAT(/,'REP. PT. INDEX WIDTH F')
N(1)=1
TW(1)=T1/2.
FFIT(1)=F(1)
DO 256 K=2,J
N(K)=K
TW(K)=T1
256 FFIT(K)=F(K)
WRITE(6,254)(N(K),TW(K),FFIT(K),K=1,J)
254 FORMAT(15,2E15.6)
N(J+1)=(JLAST+J)/2
FFITJ=0.0
JM=JLAST-1
IF(N(J+1).EQ.J)N(J+1)=J+1
IF(J.EQ.JM)GO TO 357
DO 257 J2=J+1,JM
257 FFITJ=FFITJ+F(J2)

```

```

357 FFITJ=FFITJ+F(JLAST)/2.
POINTS=FLOAT(JLAST-J)-0.5
FFIT(J+1)=FFITJ/POINTS
TW(J+1)=POINTS*T1
WRITE(6,254)N(J+1),TW(J+1),FFIT(J+1)
JMID=JLAST-JWIDE/2
DO 260 K=J+2,NFIT
JMID=JMID+JWIDE
N(K)=JMID
JFIRST=JLAST
JLAST=JFIRST+JWIDE
FFITK=(F(JFIRST)+F(JLAST))/2.
IF(JWIDE.EQ.1) GO TO 259
JP=JFIRST+1
JM=JLAST-1
DO 258 J=JP,JM
FFITK=FFITK+F(J)
258 CONTINUE
259 FFIT(K)=FFITK/FLOAT(JWIDE)
TW(K)=T1*FLOAT(JWIDE)
WRITE(6,254)N(K),TW(K),FFIT(K)
260 CONTINUE
61 RETURN
END

```

APPENDIX D. MACEDIT UTILIZATION INSTRUCTIONS

The MACEDIT code produces a normalized set of macroscopic Legendre coefficients from the values on the MAC file. All values are normalized by the factor necessary to change the first coefficient to some input value. The first value, A_0 , is the total scattering cross section and can be used as a basis for manipulating the entire set. As examples of such use, an approximate set of A_n values might be produced from an incomplete LEG file or a MIELEG run with a small number of PARTS. This may be sufficient detail to produce relative values (and determine the order necessary), but insufficient for absolute magnitude. Separate information of the magnitude of the scattering cross section (as, for instance, from a MIE-2 code run) could then produce a complete macroscopic Legendre coefficient set for use in a transport code.

D.1 MACEDIT Input Description

COEFO (E15.0)

MACEDIT input consists of a single value, COEFO, and the MAC data file. The value, COEFO, is the magnitude of the scattering cross section to which the first coefficient is normalized and thus forms the normalizing factor for all other coefficients.

D.2 MACEDIT Output Description - Printout

The MACEDIT printout consists of all of the parameters found on the MAC file as well as the unnormalized and normalized coefficients written on the MAC2 file.

D.3 MACEDIT Output Description - Files

The MACEDIT code reproduces the MAC-file coefficients on the MAC2 file, then writes the normalized coefficients on the MAC2 file

as well. Since any MACRO run or MIFLEG run (with XSOR or XSLA options) will write over the MAC file, the MAC2 file can serve as a preserve of the MAC data. (In general, any MAC or MAC2 data which is to be saved indefinitely should be copied to some other file to avoid inadvertant loss.)

D.4 MACEDIT FORTRAN Listing

As well as being a member of the MACEDIT Code of Practice, the MACEDIT Code of Practice will also cover the MACEDIT Code of Practice in the MACEDIT Code of Practice. In general, the MACEDIT Code of Practice should be applied to all other MACEDIT Code of Practice (to avoid confusion).

MACEDIT CODE OF PRACTICE

MACEDIT CODE

AD-A048 614

RADIATION RESEARCH ASSOCIATES INC FORT WORTH TEX
STUDIES IN APPLICATION OF DISCRETE ORDINATES TRANSPORT METHODS --ETC(U)
SEP 77 D LINDSTROM

F/G 20/6

F08606-77-C-0008

UNCLASSIFIED

RRA-T7704

AFTAC-TR-78-1

NL

292

ADAO48 614



END
DATE
FILMED

2 -78

DDC

C
 C MACEDIT - CODE TO NORMALIZE MACROSCOPIC LEGENDRE COEFFICIENTS ON THE
 C MAC FILE TO A KNOWN SCATTERING CROSS-SECTION (FIRST COEFFICIENT)
 C

```

COMPLEX PM
DIMENSION COEFXS(500),TITLE(20),COEFX2(500),STUFF(8)
ZERO =0.0
REWIND 15
READ(5,10)COEF0
10 FORMAT(E15.0 )
READ(15,4)(TITLE(I),I=1,20)
WRITE(16,4)(TITLE(I),I=1,20)
4 FORMAT(/,20A4)
WRITE(6,104)(TITLE(I),I=1,20),COEF0
104 FORMAT(/,'MACEDIT FOR COEFS. OF CASE:',/,20A4/
1 ' NORMALIZED TO',E15.4)
READ(15,91)RMIN,RMAX,PM,WAVEL,NEQ,P1,P2,P3,P4,NRS,LESS,EN,MORDER
91 FORMAT(/,10X,/,5E15.6,/,14,4E15.6,/,215,E15.6,I5)
WRITE(16,91)RMIN,RMAX,PM,WAVEL,NEQ,P1,P2,P3,P4,NRS,LESS,EN,MORDER
WRITE(6,92)RMIN,RMAX
92 FORMAT(/,1X,'RMIN=',E15.6,5X,'RMAX=',E15.6)
WRITE(6,93)PM,WAVEL
93 FORMAT(/,1X,'PM=',2E15.6,5X,'WAVEL=',E15.6)
WRITE(6,94)NEQ,P1,P2,P3,P4
94 FORMAT(/,1X,'NEQ=',I4,/, 'DISTRIB. PARAMETERS',5X,4E15.6)
WRITE(6,95)NRS,LESS,EN,MORDER
95 FORMAT(/,1X,'NRS=',I5,5X,'LESS=',I5,5X,'EN=',E15.6,
1 ' SUGG. P-ORDER='I5)
READ(15,200)(STUFF(I),I=1,8),DUM,DUM,COEFXS(1),DUM,
1 (DUM,DUM,DUM,COEFXS(I),I=2,LESS)
200 FORMAT(/,7A4,/,A4,/, (4E12.5))
WRITE(16,201)(STUFF(I),I=1,8),ZERO,ZERO,COEFXS(1),COEFXS(1),
1 (ZERO,ZERO,ZERO,COEFXS(I),I=2,LESS)
201 FORMAT(/,7A4,/,A4,/,2F12.1,2E12.5,/, (3F12.1,E12.5))
FACTOR = COEF0/COEFXS(1)
DO 210 I=1,LESS
COEFX2(I)=FACTOR*COEFXS(I)
210 CONTINUE
WRITE(6,220)(STUFF(I),I=1,8)
220 FORMAT(/,1X,7A4,/,1X,A4)
WRITE(6,240)(COEFXS(I),COEFX2(I),I=1,LESS)
240 FORMAT(/,1X,'COEFFICIENTS'.,/,7X,'ORIGINAL',7X,'NORMALIZED'./
1 (1X,2E15.5))
WRITE(16,250)COEF0,STUFF(8),ZERO,ZERO,COEFX2(1),COEFX2(1),
1 (ZERO,ZERO,ZERO,COEFX2(I),I=2,LESS)
250 FORMAT(/,'COEFFICIENT NORMALIZED TO',E15.6,/,A4,/,2F12.1,2E12.5,
1 /, (3F12.1,E12.5))
STOP
END

```

APPENDIX E. TDA UTILIZATION INSTRUCTIONS

(From CCC-180, Time-Dependent Multigroup One-Dimensional Discrete Ordinates Transport Code, RSIC, ORNL)

E.1 Source and Initial Conditions

Time-dependent ANISN offers a choice of two types of sources and one initial condition specification. A space and energy distributed source with a step function time distribution is available. This source is set equal to zero by the program after the first time interval. An analytic first-collision source provides an accurate representation of delta function or time dependent, point (in space), isotropic sources in spheres, infinite plane, mono-directional sources in slab geometry, and infinite line, isotropic-perpendicular-to-the-line sources in cylinders. All sources may have an arbitrary distribution in energy. The complete, centered in space and angle, flux distribution at $T=0$ may be specified alone or in addition to one of the above sources. If both a source and the initial condition are specified, the normalization factor should be zero since only one of the arrays will be normalized.

E.2 Output Edit

It is apparent that attempts to print all the information generated by a time-dependent discrete ordinates program in each time interval would be impractical and indeed useless. Time-dependent ANISN generates a tape (or disk, etc.) which contains the scalar flux, the out-going angular flux at both boundaries for each time interval, and the fission neutron density. If the analytic first-collision source option is used and the user desires, the uncollided flux is also written on this tape. In addition, the user may specify that the complete angular flux in each time interval be written on a separate tape. Subroutine EDIT is designed so that it may be easily modified to suit the demands of the user. All parameters and any ANISN data arrays which could

conceivably be used in an analysis of these data sets are available for use in the subroutine.

The version of EDIT distributed with the program will, at the users' option, print the scalar flux, the uncollided flux if available, and compute activities from user-supplied activity cross sections.

E.3 Input Specifications

All numerical data is written in the FIDO format used in ANISN. A complete description of the format and convenience options is found in Appendix A. Since familiarity with ANISN is assumed, the following data description is brief except for those parameters or arrays which are now concerned with time dependence. As in ANISN, the quantity in brackets is the array dimension and the expression in braces is the condition requiring entry of an array or set of arrays. If no condition is specified, entry of the array or set is required. A T follows each set which is entered.

A. Title card - format (12A4, 18X, I6)

A time limit in minutes may be entered in columns 67-72. The case is terminated following the first time interval in which the time limit is exceeded.

B. Parameters

15\$ Integer parameters [36]

1. ID problem identification number
2. ITH - 0 - forward solution
1 - adjoint solution
3. ISCT maximum order of Legendre polynomial approximation to scattering cross sections
4. ISN angular quadrature order
5. IGE 1 - slab
2 - cylinder
3 - sphere

6. IBL left boundary condition
 0 - vacuum
 1 - reflection
 2 - periodic
 3 - white/albedo
7. IBR right boundary condition, same options as IBL
8. IZM number of zones
9. IM number of mesh intervals
10. IEVT 0
11. IGM number of energy groups
12. IHT position of σ_{total} in cross-section table
13. IHS position of σ_{gg} in cross-section table
14. IHM length of cross-section table
15. MS length of cross-section mixing table
16. MCR number of cross-section sets to be read from cards
17. MTP number of cross-section sets to be read from library tape
18. MT total number of cross-section sets
19. IDFM 0 - no effect
 1 - enter density factors (21*)
20. IPVT 0
21. IQM 0 - no effect
 1 - enter distributed source to be used in first time interval, only
22. IPM 0 - no effect
 IM - enter complete centered angular flux distribution at T=0, (IM, MM, IGM)
23. IPP 0
24. IIM inner iteration maximum per group per outer iteration
25. ID1 0 - no effect
 1 - print scalar flux
 2 - print uncollided flux if IFG 0
 3 - both 1 and 2

- 26. ID2 0
- 27. ID3 number of time intervals
- 28. ID4 number of activities
- 29. ICM outer iteration maximum
- 30. IDAT1 0
- 31. IDAT2 0
- 32. IFG 0 - no effect
N - enter N source spectra (23*) for first-collision source calculation
- 33. IFLU no. of time steps for first-collision source
- 34. IFN 1 - enter flux guess (3*)
- 35. IPRT 0 - no effect
1 - do not print cross sections
- 36. IXTR 0 - no effect
1 - write angular flux tape for each time interval

16* Floating point parameters [14]

- 1. EV 0.0
- 2. EVM 0.0
- 3. EPS accuracy desired
- 4. BF buckling factor, normally 1.420892
- 5. DY cylinder
- 6. DZ plane depth for buckling correction
- 7. DFML transverse dimension for void streaming correction
- 8. XNF normalization factor
- 9. PV 0.0
- 10. RYF 0.5
- 11. XLAL point flux convergence criterion
- 12. XLAH 0.0
- 13. EQL 0.0
- 14. XNPM 0.0

T

C. Cross Sections

13\$ cross-section library ID numbers [MTP] {MTP>0}

14* MCR cross-section sets [IHM*IGM*MCR] {MCR>0}

T

D. Distributed source-centered flux distribution {IQM=1 or IPM=IM}

17* distributed source [IM*IGM] {IQM=1}

18* centered angular flux distribution [IM*MM*IGM] {IPM=IM}

T

E. Flux guess {IFN=1}

3* flux guess [IM*IGM] {IFN=1}

T

F. Remainder of data

1* fission spectrum [IGM]

4* radii [IM+1]

5* velocities [IGM]

6* angular quadrature weights [MM]

7* angular quadrature cosines [MM]

8\$ zone numbers [IM]

9\$ material numbers [IZM]

10\$ mixture numbers [MS]

11\$ component numbers [MS]

12* number of densities [MS]

19\$ order of Legendre polynomial approximation to scattering cross section [IZM]

20* activity cross sections [IGM*ID4] {ID4>0}

21* density factors [IM] {IDFM=1}

22* time interval boundaries [ID3+1]

23* IFG source spectra; source emission occurs at corresponding times specified by 24* array; the first spectrum is normalized to XNF (if XNF>0) and the other spectra are scaled to maintain the same relative values [IGM*IFG]{IFG>0}

- 24* source emission time interval boundaries for first-collision source spectra [IFLU+1] {IFG>0}
- 25* right boundary albedo [IGM] {IBR=3}
- 26* left boundary albedo [IGM] {IBL=3}
- 27\$ source spectrum no. by time step [IFLU]

T

E.4 Data Array Formats

All input data, with the exception of the title card, are read using the same format.

E.4.1 Type 1 Format

Each card is divided up into six 12-digit data fields which are in turn divided up into three subfields, illustrated in the following figure. Only one data field is shown.



The first subfield is a two-digit integer; the second subfield contains either a \$, *, R, I, T, S, F, A, C, E, Q, L, N, M, \emptyset , U, V, Z, +, -, or a blank. The third subfield contains either a fixed or floating point number. The contents of the first two subfields will define the operation to be performed on the third field.

Blank fields are ignored. One can use any or all fields on a card. For example, a box of blank cards sandwiched anywhere in a data array would be completely ignored.

Each data array is identified by a two-digit integer in a first subfield. There are both fixed and floating point arrays; a fixed point

array is designated by a \$ in the second subfield, a floating point array by an *.

The second subfield contains an operator which specifies the type of operation to be performed on the data. The possible operators are listed below.

E.4.2 Array Operators

\$ indicates the beginning of an integer array. The first subfield contains a one- or two-digit number identifying the array.

* indicates the beginning of a floating point array. The first subfield identifies the array.

R indicates that the entry in the third subfield is to be repeated the number of times specified in the first subfield.

I indicates linear interpolation between the entry in the third subfield and the entry in the third subfield of the next data field. The number in the first subfield gives the number of points to be placed equally spaced in the specified range.

T indicates termination of data reading for a block. XLACS can require several data blocks and each block must be ended with a T. A block can contain any number of arrays. Data on a card after a T will be ignored.

S indicates skip. The first subfield defines the number of entries to be skipped. The third field can contain the first entry following the skips. A blank third subfield would be ignored.

F is used to fill the remainder of an array with the item given in the third subfield.

A is used to address a particular location in an array. This location is specified in the third subfield; the first subfield is blank.

C is used to obtain a count of the number of items read into an array up to the point where the C is placed. An integer ZZ in front of the C will be used as identification in producing a message as follows:

XX ENTRIES READ IN THE YY ARRAY at ZZC.

E may be used to end specifying data for an array. This option is particularly useful when it is desired to replace only some items in a particular array. The items in question are replaced, and the use of an E prevents having to count and skip to the end of the array.

Q is used to repeat sequences of numbers. The length of the sequence is defined in the third subfield. The number of times to repeat the sequence is given in the first subfield.

L is similar to I except that a logarithmic interpolation is performed between the entry points. This option is particularly useful for defining energy structures equally spaced in lethargy.

N is used to repeat a sequence of numbers in reverse order. The length of the sequence is defined in the third subfield. The number of times to repeat a sequence is given in the first subfield.

M is used to negate and repeat an inverted sequence. The length of the sequence is given in the third subfield. The number of times to repeat a sequence is given in the first subfield.

Ø is used to turn on (or off) the card image edit of ANISN input data. As with the C option, an integer in front of the Ø identifies the particular entry. The default (starting) condition is not to edit the data.

U is used to replace the ANISN input format for an array. The array number is given in the first subfield. The format, written in normal FORTRAN, is specified on the card immediately following the card containing a U. The parentheses normally capsulating a format should be included.

V specifies that the array identified in the first subfield will be read according to the last variable format read in.

Z is used to specify a string of zeros; e.g., 49Z would place forty-nine zeros into an array.

+ or - indicates exponentiation. The data in the third field is multiplied by $10^{\pm N}$, where N is an integer in the first subfield. This option allows one to specify a number in up to nine significant digits.

Integer data in the third subfield must be right adjusted. Floating point data may be written with or without an exponent. If the decimal is omitted, it is assumed to be immediately to the left of the exponent field. If there is no exponent, the decimal point is assumed to be to the extreme right of the nine-column subfield.

E.4.3 Input Restrictions

The following restrictions must be observed when using the ANISN input format:

- (1) Blank data fields are ignored.
- (2) If the interpolation option (I) is used, the next data field may not be either blank or an A entry.
- (3) The third subfield of a data field containing a \$ or an * may contain an integer, N. The next data entry is assumed to be the (N+1)th member of the array. Normally the third subfield is a blank and is ignored.
- (4) All data arrays must be filled with the correct number of entries. A data array is ended by either starting a new data array or by ending a data block.

E.4.4 Type 2 Format (Free Form)

The transferral of input data to input forms or punched cards for a code requiring significant amounts of input is always a time consuming, distasteful, and error-prone process. The original ANISN formats were designed to help reduce these difficulties. The options are convenience features. The usefulness of the "F" option which fills an array is obvious, but it is somewhat harder to see the practical uses for some of the more obscure ones like N, M and Q; however, frequent use will turn up situations where these options are invaluable. For example, the S_n cosines are negated and reflected about 90° , a fact which suggests the use of the M option.

There are justifiable complaints with the input formats; for example, where convenience options are not applicable, data can be hard to write because of the manner in which the data fields are spread on the card. This is especially true of integer arrays, where the data are right adjusted in 12-column fields. The ANISN input forms help to some extent, but the actual keypunching is still troublesome.

The input format has been greatly improved by Ward Engle of ORNL who has designed and implemented an all-FORTRAN free-form, ANISN input scheme which has data items separated by blanks (as others do), but still allows all of the important convenience features of the earlier formats. The restrictions on the use of this input are essentially that the user write the data in a form that he can interpret within the context of the ANISN options. Data is easily written and keypunched, since there is no worry about which type character falls in which columns or how many blanks are left between entries.

The free-form input can be interspersed with the fixed form input. To select free-form, an array is identified as either a \$\$ or a ** array, for integer and floating point arrays, respectively.

The restrictions are:

- (1) Any third subfield (data entry) must be followed by one

or more blanks. This is an obvious restriction, otherwise data interpretation would be impossible.

(2) Only columns 1-72 are used.

(3) Numbers with exponents must not have imbedded blanks; e.g., use 1.0E+4, not 1.0 E+4 or 1.0E+ 4.

(4) The old + or - options (2nd subfield) are not operational. Significance requirements which led to the development of this option can be had directly.

(5) No more than 9 digits in a number can be entered. The exponent is not counted; e.g., 9234+09, 923400000+1 will work, 9234000000 will not work. Nine-digit accuracy is clearly beyond the significance available for single precision IBM 360 floating point operations.

(6) A blank must not appear between items which fall in the first and second subfields with the old format, e.g., 24R, not 24 R. Note that the 99 restriction on the number of repeats, interpolations, etc., has been eliminated.

(7) The Z-entry must be entered as 738Z, not Z738. The old format allowed either.

(8) The Q, M, N entries must be specified as Q4, not 4Q. The old format allows either. An entry like 3Q4 accomplishes the same as Q4 Q4 Q4. This is now true for either format.

The character (') in column 1 of a card will cause the contents of the card to be listed as comments, while the data is read in. Column 2 should contain the proper carriage control character; e.g., blank,0,1,2, etc. This card is ignored as a data card. This option is also available with the old formats.

Some examples of the new format are given below:

1\$\$25R1_0_4_3Q3_2\$\$3R42_E_T

The first 25 entries of the 1\$ array are 1's followed by 0 and 4 and

then data input to the array ends. The T terminates a data block.

```
42**_0.0_0.1666667_0.3333333_N2
```

```
43**_-1.0_-0.8819171_0.3333333 M2
```

This example puts 0.0, 0.1666667, 0.3333333, 0.333333, 0.1666667 in the 42* array and -1.0, -0.8819171, -0.3333333, 0.3333333, 0.8819171 in the 43* array.

TDA .5KM R CLOUD, .05-6.05(MACRO),P95,S48(G)

40

15 81777 0 95 48 3 3 0 1 27 0 1 3 4 4 0 96 0 96 1 0 0 0 0 50 1 0 39

0 1 0 0 1 1 1 0 0

16 2R0.0 .0001 6R0.0 0.5 .0001 3R0.0 T

14**

.0	.0	.17213E-03	.17213E-03
.0	.0	.0	.43553E-03
.0	.0	.0	.66264E-03
.0	.0	.0	.77716E-03
.0	.0	.0	.89232E-03
.0	.0	.0	.99274E-03
.0	.0	.0	.10876E-02
.0	.0	.0	.11974E-02
.0	.0	.0	.13072E-02
.0	.0	.0	.14100E-02
.0	.0	.0	.15331E-02
.0	.0	.0	.16179E-02
.0	.0	.0	.17311E-02
.0	.0	.0	.18117E-02
.0	.0	.0	.19017E-02
.0	.0	.0	.19821E-02
.0	.0	.0	.20535E-02
.0	.0	.0	.21241E-02
.0	.0	.0	.21952E-02
.0	.0	.0	.22539E-02
.0	.0	.0	.23201E-02
.0	.0	.0	.23698E-02
.0	.0	.0	.24294E-02
.0	.0	.0	.24708E-02
.0	.0	.0	.25162E-02
.0	.0	.0	.25520E-02
.0	.0	.0	.25810E-02
.0	.0	.0	.26074E-02
.0	.0	.0	.26260E-02
.0	.0	.0	.26381E-02
.0	.0	.0	.26464E-02
.0	.0	.0	.26506E-02
.0	.0	.0	.26486E-02
.0	.0	.0	.26389E-02
.0	.0	.0	.26305E-02
.0	.0	.0	.26037E-02
.0	.0	.0	.25876E-02
.0	.0	.0	.25558E-02
.0	.0	.0	.25269E-02
.0	.0	.0	.25374E-02
.0	.0	.0	.25055E-02
.0	.0	.0	.25022E-02
.0	.0	.0	.25278E-02
.0	.0	.0	.24955E-02
.0	.0	.0	.24597E-02
.0	.0	.0	.24186E-02
.0	.0	.0	.23692E-02
.0	.0	.0	.23098E-02
.0	.0	.0	.23090E-02
.0	.0	.0	.23304E-02
.0	.0	.0	.22702E-02
.0	.0	.0	.22270E-02

.0	.0	.0	.21808E-02
.0	.0	.0	.21517E-02
.0	.0	.0	.21130E-02
.0	.0	.0	.20661E-02
.0	.0	.0	.19921E-02
.0	.0	.0	.19208E-02
.0	.0	.0	.19507E-02
.0	.0	.0	.19713E-02
.0	.0	.0	.18605E-02
.0	.0	.0	.17524E-02
.0	.0	.0	.16621E-02
.0	.0	.0	.15724E-02
.0	.0	.0	.14884E-02
.0	.0	.0	.14103E-02
.0	.0	.0	.13141E-02
.0	.0	.0	.12306E-02
.0	.0	.0	.11379E-02
.0	.0	.0	.11784E-02
.0	.0	.0	.10839E-02
.0	.0	.0	.10018E-02
.0	.0	.0	.94927E-03
.0	.0	.0	.92530E-03
.0	.0	.0	.91803E-03
.0	.0	.0	.93122E-03
.0	.0	.0	.92100E-03
.0	.0	.0	.90426E-03
.0	.0	.0	.84065E-03
.0	.0	.0	.74183E-03
.0	.0	.0	.63671E-03
.0	.0	.0	.55445E-03
.0	.0	.0	.41323E-03
.0	.0	.0	.34449E-03
.0	.0	.0	.30543E-03
.0	.0	.0	.33228E-03
.0	.0	.0	.34548E-03
.0	.0	.0	.33528E-03
.0	.0	.0	.28019E-03
.0	.0	.0	.19680E-03
.0	.0	.0	.14359E-03
.0	.0	.0	.74772E-04
.0	.0	.0	.40869E-04
.0	.0	.0	.16026E-04
.0	.0	.0	.74568E-05
.0	.0	.0	-.62533E-05

T
T

3** F0.0 T
1** 1.0
4** 2610.0 5.0+04
5** 3.0+10
6**

0.0	.001577	.003664	.005739	.007790	.009808	.011785	.013713
.015588	.017389	.019121	.020773	.022337	.023808	.025180	
.026445	.027600	.028639	.029557	.030352	.031020	.031551	
.031962	.032233	.032369					
.032369	.032233	.031962	.031551	.031020	.030352	.029557	
.028639	.027600	.026445	.025180	.023808	.022337	.020773	

.019121 .017389 .015588 .013713 .011785 .009808 .007790

.005739 .003664 .001577

7**

-.999476 -.998771 -.993530 -.984125 -.970592 -.952988

-.931387 -.905879 -.876572 -.843588 -.807066 -.767159

-.724034 -.677872 -.628867 -.577225 -.523161 -.466903

-.408686 -.348756 -.287362 -.224764 -.161222 -.097005

-.032380

.032380 .097005 .161222 .224764 .287362 .348756 .408686

.466903 .523161 .577225 .628867 .677872 .724034 .767159

.807066 .843588 .876572 .905879 .931387 .952988 .970592

.984125 .993530 .998771

8\$\$ F1

9\$\$ 1

19\$\$ 70

21** 27R1.0

22** 2610.0 1.666666-6 1.8453-6 2.2024-6 2.7381-6 3.4524-6

4.3453-6 5.4167-6 6.6667-6 8.0953-6 9.7025-6 1.1488-5 1.3452-5 1.5595-5

23** 1.0

24** 0.0 0.0

26** 1.0

27\$\$ 1 T T

APPENDIX F. FILE NAME DIRECTORY

MIELEG - FORTRAN source for MIELEG Code

LM:MIELEG - MIELEG load module

MIELEGM - Job module to compile, load MIELEG

MIELEGR - Job module to run MIELEG, including data

LEGEDIT - FORTRAN source for LEGEDIT Code

LM:LEGEDIT - LEGEDIT load module

LEGEDITM - Job module to compile, load LEGEDIT

LEGEDITR - Job module to run LEGEDIT, including data

MACRO - FORTRAN source for MACRO Code

LM:MACRO - MACRO load module

MACROM - Job module to compile, load MACRO

MACROR - Job module to run MACRO, including data

MACEDIT - FORTRAN source for MACEDIT Code

LM:MACEDIT - MACEDIT load module

MACEDITM - Job module to compile, load MACEDIT

MACEDITR - Job module to run MACEDIT, including data

LEG - Microscopic Legendre coefficient data file

BACKLEG - LEG backup file

NEWLEG - LEG production file

OTHERLEG - LEG overflow file for other refractive indices

MAC - Macroscopic Legendre coefficient data file, used to produce TDA data files by MACRO.

MAC2 - Data file produced by MACEDIT from data found on MAC.

TDA - Load module for Time-Dependent ANISN jobs requiring COMMON storage of 20K or less
 TDA30K - Load module for jobs requiring 30K or less.
 TDA-IN - TDA job data file
 TDARUN - TDA run module
 TDAMAIN - TDA MAIN routine, specifies storage in COMMON and LIM1
 COMPFILE - Job module to compile TDAMAIN
 EDITBO, TDABO, GUTSBO - Other modules making up TDA (FORTRAN sources: EDIT, TDASO, and GUTS on tape LT#R104/.9002007E)
 TDAFILE - Job module to load TDA

LMANISN - Load module for steady-state ANISN (present version will not accept free-form data)
 ANISNDATA - ANISN job data file
 RUNANISN - ANISN run module

QUADS - Angular quadrature data sets in free-form arrays (6** and 7**).
 Used to produce TDA data files.

DISTRIBUTION

<u>No. of Copies</u>		<u>No. of Copies</u>	
	<u>DOD AGENCIES</u>	3	US Army Electronics Command Atmospheric Sciences Laboratory ATTN: Dr M. L. Vatsia Dr Jerry Lentz Dr Richard Gomez
2	Director Defense Nuclear Agency ATTN: RAAE Technical Library		
1	Director Defense Intelligence Agency ATTN: Technical Library		<u>DEPARTMENT OF THE NAVY</u>
2	Defense Documentation Center ATTN: Document Control	1	Naval Research Laboratory ATTN: Technical Library
			<u>DEPARTMENT OF THE AIR FORCE</u>
	<u>ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION</u>	3	AF Geophysics Laboratories ATTN: OPR (Mr H. Gardiner) OPA (Dr R. Fenn) OPR (Dr A. T. Stair)
3	Los Alamos Scientific Laboratory ATTN: J-10 (Dr H. Horak) J-10 (Dr Guy Barasch) TD-3 (Dr Bob Henson)	1	AF Institute of Technology ATTN: Library
3	Sandia Laboratories ATTN: Bob Bradley Gary Mauth Dean Thornbrough	2	AF Weapons Laboratory ATTN: Dr Joe Janni Dr C. Needham
1	Lawrence Livermore Laboratory ATTN: Technical Library	1	Space and Missile Systems Organization ATTN: SZS (Maj H. Hayden)
2	Oak Ridge National Laboratory ATTN: Technical Library RSIC	2	AF Technical Applications Center ATTN: TFR (Capt J. Lange)
1	EG&G, Inc. Los Alamos NM ATTN: Dr C. Mitchell		<u>DOD CONTRACTORS</u>
	<u>DEPARTMENT OF THE ARMY</u>	1	Aerospace Corporation ATTN: Dr R. D. Rawcliffe
2	US Army Waterways Experiment Station-Mobility and Environmental System Lab ATTN: Dr L. E. Link Dr Warren Grabau	1	General Electric Company-TEMPO ATTN: DNA Information and Analysis Center

DISTRIBUTION (Continued)

No. of
Copies

- 1 Lockheed Missiles and Space
Company
ATTN: Technical Library
- 1 Visidyne, Inc.
ATTN: Dr J. Carpenter
- 1 Mission Research Corporation
ATTN: Technical Library
- 1 Science Applications, Inc.
ATTN: Dr R. Hillendahl
- 1 HSS, Inc.
ATTN: Dr H. Stewart
- 20 Radiation Research Associates
ATTN: M. B. Wells