ARGONNE NATIONAL LABORATORY

ONE-DIMENSIONAL TRANSPORT CODE FOR ONE-GROUP PROBLEMS IN PLANE GEOMETRY

by

Erwin H. Bareiss and Cynthia Chamot

MASTER

U of C-AUA-USAEC

**APPLIED
MATHEMATICS
DIVISION**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona	Kansas State University	The Ohio State University
Carnegie-Mellon University	The University of Kansas	Ohio University
Case Western Reserve University	Loyola University	The Pennsylvania State University
The University of Chicago	Marquette University	Purdue University
University of Cincinnati	Michigan State University	Saint Louis University
Illinois Institute of Technology	The University of Michigan	Southern Illinois University
University of Illinois	University of Minnesota	The University of Texas at Austin
Indiana University	University of Missouri	Washington University
Iowa State University	Northwestern University	Wayne State University
The University of Iowa	University of Notre Dame	The University of Wisconsin

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

ARGONNE NATIONAL LABORATORY
Argonne, Illinois 60439
Applied Mathematics Division

ONE-DIMENSIONAL TRANSPORT CODE FOR ONE-GROUP
PROBLEMS IN PLANE GEOMETRY

Erwin H. Bareiss and Cynthia Chamot

September 1970

Technical Memorandum No. 209

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed; or represents that its use would not infringe privately owned rights.

This report was prepared primarily for internal use at Argonne National Laboratory and may contain preliminary or unvarified information.

* Work performed under the auspices of the U. S. Atomic Energy Commission.

This memorandum was written in October 1962. It is re-issued because it contains considerable computational experience with Hermite Interpolation, which is considered of prime importance for the construction of an efficient analytical transport code in three-space dimensions.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

for

Table of Contents

Foreword	-----	i
Abstract	-----	1
I. Introduction	-----	1
II. Numerical Analysis and Results	-----	2
A. Iteration Procedures	-----	2
B. Direct Matrix Solution	-----	4
III. Description of Computer Program 790/RE204	-----	13
Appendix A: Integration Constants	-----	18
Appendix B: Schechter's Method for Solving Quasi-Tridiagonal Matrices	-----	20
Appendix C: FORTRAN Listing	-----	21
Appendix D: Sample Problem	-----	28
Appendix E: Tables to Compare Speed and Accuracy of Various Methods of Solution	-----	32

Foreword

This report was written in 1962 as a memorandum for file. It summarizes those results, obtained by numerical experimentation for the approximate solution of the one-dimensional stationary transport equation, based on piecewise polynomial approximations both in space and angular flux distribution. The selected approximations were judged to give superior answers to the methods currently in use and they could serve as a base for the design of multidimensional transport codes. According to a theoretical prediction, the Quade-Hermite integration (a special case that can be derived from the Birkhoff-Hermite interpolation) gave results that promise an improvement of a factor exceeding ten in computational efficiency and savings in storage requirements. Essentially two formulas were selected for extensive study and comparison; the first order method carries only first-order terms in space direction, but has a truncation error of order three, while the second order method carries second order terms which result in a truncation error of order five with a very small multiplication factor (1/720).

The report is re-issued because the method of piecewise approximation has recently obtained prominence in the "method of finite" elements in the numerical solution of partial differential equations, and the 15 tables of numerical results will provide the much needed "numerical" support to the theory, especially with respect to applications in transport theory. A brief discussion of the theoretical

background of the methods investigated in the project leading to this report is given in E. H. Bareiss "A Survey and Classification of Transport Theory Calculations Techniques," Proc. United Nations Int'l. Conf. on Peaceful Uses of Atomic Energy, Vol. 16, pp. 503-516, Geneva, 1958. An extensive treatment on optimum truncation errors and numerical stability is given in John H. Bennett "Truncation Errors in Numerical Solutions of the Transport Equation," Thesis, Harvard University, 1962.

ONE-DIMENSIONAL TRANSPORT CODE FOR ONE-GROUP
PROBLEMS IN PLANE GEOMETRY

by

Erwin H. Bareiss and Cynthia Chamot

ABSTRACT

Equations and results are given for various methods of solution of the one-dimensional transport equation for one energy group in plane geometry with inelastic scattering and an isotropic source. After considerable investigation, a matrix method of solution was found to be faster and more stable than iteration procedures. A description of the code is included which allows for up to 24 regions, 250 points, and 16 angles such that the product of the number of angles and the number of points is less than 600.

I. INTRODUCTION

This report describes the work done to determine the optimum way to do calculations to obtain accurate results fast enough to be useful in more general reactor problems. Different procedures based on the transport equation were used to calculate angular fluxes for various combinations of points and angles. For all codes, the integration for the scalar flux calculation used either Single Gauss, or Double Gauss, or Midpoint formulas. The results given in the various tables are for a sample problem with two regions, Uranium and water.

II. NUMERICAL ANALYSIS AND RESULTS

A. Iteration Procedures

Initially two iteration schemes were coded with a first-order ($\epsilon=0$) and a second order ($\epsilon=1$) approximation available. The basic equation was

$$F_{\lambda j}^{i+1} = (1 + C_{\lambda r})^{-1} \left[F_{\lambda j-1}^i + \frac{\Delta x_r}{2} \left(\frac{\partial F_{\lambda j}^i}{\partial x} + \frac{\partial F_{\lambda j-1}^i}{\partial x} \right) - \epsilon \frac{\Delta x_r^2}{12} \left(\frac{\partial^2 F_{\lambda j}^i}{\partial x^2} - \frac{\partial^2 F_{\lambda j-1}^i}{\partial x^2} \right) + C_{\lambda r} F_{\lambda j}^i \right].$$

Expressions for 1st and 2nd derivatives are the same as those described under B for the direct matrix solution. For the first scheme, $C_{\lambda r} = \frac{\Delta x_r}{2\mu_\lambda} \sum_r^t$. About 100 iterations were needed to obtain five significant figures (see Tables 2-6). There was considerable instability in the second degree solutions attempted. Results were unobtainable when there were too few points for the number of angles, and also when there were too few angles for the number of points (see Table 1). Instability first is indicated as cases take more and more iterations to converge.

For the second scheme, $C_{\lambda r} = \frac{\Delta x_r}{2\mu_\lambda} \left(\sum_r^t - \frac{1}{2} \sum_r^s R_\lambda \right) + \frac{\epsilon}{3} \left(\frac{\Delta x_r}{2\mu_\lambda} \right)^2 \sum_r^t \left[\sum_r^t - \sum_r^s R_\lambda \right]$. This method of solution did not reduce the number of iterations necessary, and also exhibited instabilities (see Tables 1-7).

Nevertheless, accurate values of the scalar fluxes were obtained, using different numbers of points and angles (see Tables 8-11). Breaking each region into four intervals gives the same results to five figures as with many more points as long as a second degree approximation formula is used. The first-order approximation gave poor results, especially for too

few points (see Tables 8-10). Though the Double Gauss solution presented more stability problems (see Table 1), the scalar fluxes calculated for few angles agreed with the values obtained for many angles, better than with the other integration formulas. This may be seen by studying the information in Table 15.

Thus, the optimum solution appears to be based on a Double Gauss integration for the scalar flux, using six angles and four points in each region, with a second degree approximation equation. The time involved in this calculation is thus the major concern now, and may be determined as follows for the iteration scheme:

$$\text{No. of seconds} = \frac{(\text{No. of angles}) * (\text{No. of Points}) * (\text{No. of Iterations})}{80}$$

This gives a little over a minute for the suggested mesh size and the usual 100 iterations. Some variations in the iteration procedure were tried, but the overall machine time could not be reduced appreciably.

- i) Starting the fluxes at some value other than zero had only a slight effect on the number of iterations necessary to obtain the desired accuracy (see Table 13 in Appendix E). Also, Table 7 shows that starting with approximately correct values seldom cuts the number of iterations by more than half.
- ii) Starting with a rough mesh initially, and then interpolating the fluxes at points or angles in between, and iterating for successively finer meshes, gives little assurance that machine time will be saved. In fact, if the fluxes are determined to full accuracy for each mesh, the calculation might take more

than twice as long as the usual method of running the case with only the fine mesh (see Table 7 in Appendix E).

- iii) Extrapolating after iterating to about 5% accuracy is sometimes possible, but was not in general satisfactory because cases may be unstable.
- iv) The results for various ratios of $\sum^s$ to $\sum^t$ are given in Table 12 in Appendix E.
- v) It may be seen from studying Tables 2-6 that about the same number of iterations are necessary for each type of integration formula used to calculate the scalar fluxes.

B. Direct Matrix Solution

A second order Quade-Hermite Integration formula was used in conjunction with the transport equation and boundary conditions, to generate a system of linear equations whose solutions are the angular fluxes. A simple elimination method originally used to solve the system, fails in certain cases (see Table 14), unless double precision is used. However, since the matrix is of a Quasi-Tridiagonal form (see Figure 1), it may be solved satisfactorily in single precision by Schechter's scheme for such matrices.* See Appendix B for a brief summary.

Then, as before, scalar fluxes are calculated using either of the three integration formulas.

* NYO-2542: Quasi-Tridiagonal Matrices by Samuel Schechter of New York University (May 1, 1959).

The running time for the matrix method of solution is much less than that for the iteration method, and may be calculated as follows:

$$(\text{No. of seconds}) = (\text{No. of angles})^2 \cdot (\text{No. of points}) \div 40.$$

This gives seven seconds for the suggested mesh size, compared with more than a minute by iterating.

Transport Equation

The one-dimensional, one-velocity transport equation for plane (slab) geometry is

$$\mu \cdot \frac{\partial F(x, \mu)}{\partial x} + \sum^t(x) \cdot F(x, \mu) = \sum^s(x) \int_{-1}^1 P(x, \mu, \mu') F(x, \mu') d\mu' + S(x, \mu).$$

If the sources are isotropic, S is a function of x only. If, in addition, inelastic scattering is assumed, then P is independent of μ and x and μ' , and is always equal to $1/2$. The transport equation then simplifies to

$$\mu \cdot \frac{\partial F(x, \mu)}{\partial x} + F(x, \mu) \cdot \sum^t = \frac{1}{2} S(x) + \sum^s/2 \cdot \phi(x)$$

where

$$\phi(x) = \int_{-1}^1 F(x, \mu') d\mu'.$$

Rearranging terms and setting the scalar flux for the j th point along x ,

ϕ_j , equal to $\sum_{\text{all } \lambda} R_{\lambda} F_{\lambda j}$, gives an expression for the first derivative.

$$F'_{\lambda j} = \frac{\partial F_{\lambda j}}{\partial x} = - \frac{\sum^t}{\mu_\lambda} \cdot F_{\lambda j} + \frac{\sum^s}{2\mu_\lambda} \sum_{\text{all } \lambda'} R_{\lambda'} F_{\lambda' j} + \frac{1}{2} S/\mu_\lambda$$

where μ_λ and R_λ are the coefficients and weights at appropriate angles for the type of integration. This simplifies to

$$F'_{\lambda j} = a_\lambda R_\lambda F_{\lambda j} + b_\lambda \sum_{\text{all } \lambda'} R_{\lambda'} F_{\lambda' j} + c_\lambda$$

where

$$a_\lambda = - \sum^t / (\mu_\lambda R_\lambda)$$

$$b_\lambda = \sum^s / (2\mu_\lambda)$$

$$c_\lambda = \frac{1}{2} S/\mu_\lambda.$$

An expression for the second derivative is then derived as follows:

$$F''_{\lambda j} = \frac{\partial^2 F_{\lambda j}}{\partial x^2} = a_\lambda R_\lambda \frac{\partial F_{\lambda j}}{\partial x} + b_\lambda \sum_{\text{all } \lambda'} R_{\lambda'} \frac{\partial F_{\lambda' j}}{\partial x}$$

Substituting for the first derivatives gives

$$\begin{aligned} F''_j &= a_\lambda R_\lambda a_\lambda R_\lambda F_{\lambda j} + a_\lambda R_\lambda b_\lambda \sum_{\text{all } \lambda'} R_{\lambda'} F_{\lambda' j} + a_\lambda R_\lambda c_\lambda \\ &+ b_\lambda \sum_{\text{all } \lambda'} R_{\lambda'} (a_\lambda R_\lambda F_{\lambda' j}) + \underline{b_\lambda \sum_{\text{all } \lambda'} R_{\lambda'} (b_{\lambda'} \sum_{\text{all } \lambda''} R_{\lambda''} F_{\lambda'' j})} \\ &+ b_\lambda \sum_{\text{all } \lambda'} R_{\lambda'} (c_{\lambda'}). \end{aligned}$$

The underlined term drops out for our case, since for each positive $\mu_{\lambda'}$, there is a negative μ_{λ} , and each is associated with the same value for R_{λ} . This may be seen more easily if the term is rewritten:

$b_{\lambda} \sum_{\text{all } \lambda'} [R_{\lambda'} \sum^S / (2\mu_{\lambda'})] \cdot \phi_j$, where b_{λ} has been replaced by its equivalent, and $\sum_{\text{all } \lambda''} R_{\lambda''} F_{\lambda''j}$ has been replaced by ϕ_j . Thus after regrouping the terms, the second derivative may be written

$$\begin{aligned} F''_{\lambda j} &= (a_{\lambda} R_{\lambda} a_{\lambda} + a_{\lambda} R_{\lambda} b_{\lambda} + b_{\lambda} R_{\lambda} a_{\lambda}) R_{\lambda} F_{\lambda j} \\ &+ \sum_{\lambda' \neq \lambda} (a_{\lambda} R_{\lambda} b_{\lambda'} + b_{\lambda} R_{\lambda'} a_{\lambda}) R_{\lambda'} F_{\lambda' j} \\ &+ a_{\lambda} R_{\lambda} c_{\lambda} + b_{\lambda} \sum_{\text{all } \lambda'} R_{\lambda'} c_{\lambda'} \end{aligned}$$

This simplifies to

$$F''_{\lambda j} = d_{\lambda} R_{\lambda} F_{\lambda j} + \sum_{\lambda' \neq \lambda} e_{\lambda \lambda'} R_{\lambda'} F_{\lambda' j} + h_{\lambda}$$

where

$$\begin{cases} d_{\lambda} = a_{\lambda} R_{\lambda} (a_{\lambda} + 2b_{\lambda}) \\ e_{\lambda \lambda'} = b_{\lambda} (a_{\lambda} R_{\lambda'} + a_{\lambda'} R_{\lambda}) \\ h_{\lambda} = a_{\lambda} R_{\lambda} c_{\lambda} + b_{\lambda} \sum_{\text{all } \lambda'} R_{\lambda'} c_{\lambda'} \end{cases}$$

It should be noted that $e_{\lambda \lambda'} = 0$ when $\mu_{\lambda} = -\mu_{\lambda'}$.

Quade Hermite Integration

The expressions for the first and second derivatives of the flux are then inserted in the Quade-Hermite Integration Formula as follows:

$$F_{\lambda j+1} - F_{\lambda j} = \frac{\Delta x}{2} (F'_{\lambda j+1} + F'_{\lambda j}) - \frac{\Delta x}{12} (F''_{\lambda j+1} - F''_{\lambda j})$$

$$\begin{aligned} \frac{1}{R_\lambda} (R_\lambda F_{\lambda j+1} - R_\lambda F_{\lambda j}) &= \frac{\Delta x}{2} [(a_\lambda + b_\lambda) R_\lambda F_{\lambda j+1} + b_\lambda \sum_{\lambda' \neq \lambda} R_{\lambda'} F_{\lambda' j+1} + c_\lambda \\ &\quad + (a_\lambda + b_\lambda) R_\lambda F_{\lambda j} + b_\lambda \sum_{\lambda' \neq \lambda} R_{\lambda'} F_{\lambda' j} + e_\lambda] \\ &\quad - \frac{\Delta x^2}{12} [d_\lambda R_\lambda F_{\lambda j+1} + \sum_{\lambda' \neq \lambda} e_{\lambda\lambda'} R_{\lambda'} F_{\lambda' j+1} + h_\lambda \\ &\quad - (d_\lambda R_\lambda F_{\lambda j} + \sum_{\lambda' \neq \lambda} e_{\lambda\lambda'} R_{\lambda'} F_{\lambda' j} + h_\lambda)]. \end{aligned}$$

Rearranging and substituting $f_{\lambda j}$ for $(R_\lambda F_{\lambda j})$ gives

$$\begin{aligned} f_{\lambda j+1} - f_{\lambda j} &= R_\lambda \frac{\Delta x}{2} [(a_\lambda + b_\lambda - \frac{\Delta x}{6} d_\lambda) f_{\lambda j+1} + \sum_{\lambda' \neq \lambda} (b_\lambda - \frac{\Delta x}{6} e_{\lambda\lambda'}) f_{\lambda' j+1} \\ &\quad + (a_\lambda + b_\lambda + \frac{\Delta x}{6} d_\lambda) f_{\lambda j} + \sum_{\lambda' \neq \lambda} (b_\lambda + \frac{\Delta x}{6} e_{\lambda\lambda'}) f_{\lambda' j} + 2c_\lambda]. \end{aligned}$$

This simplifies to the following:

$$\begin{aligned} f_{\lambda j+1} - f_{\lambda j} &= (\alpha_\lambda + \gamma_\lambda) f_{\lambda j+1} + \sum_{\lambda' \neq \lambda} (\beta_\lambda + \epsilon_{\lambda\lambda'}) f_{\lambda' j+1} \\ &\quad + R_\lambda \Delta x c_\lambda + (\alpha_\lambda - \gamma_\lambda) f_{\lambda j} + \sum_{\lambda' \neq \lambda} (\beta_\lambda - \epsilon_{\lambda\lambda'}) f_{\lambda' j} \end{aligned}$$

where

$$\left\{ \begin{array}{l} \alpha_{\lambda} = R_{\lambda} \frac{\Delta x}{2} (a_{\lambda} + b_{\lambda}) \\ \beta_{\lambda} = R_{\lambda} \frac{\Delta x}{2} b_{\lambda} \\ \gamma_{\lambda} = - R_{\lambda} \frac{\Delta x^2}{12} d_{\lambda} \\ \epsilon_{\lambda\lambda'} = - R_{\lambda} \frac{\Delta x^2}{12} e_{\lambda\lambda'} \end{array} \right.$$

Note that $\epsilon_{\lambda\lambda'} = 0$ when $\mu_{\lambda} = -\mu_{\lambda'}$.

Collecting terms and eliminating $f_{\lambda j}$ and c_{λ} gives

$$\begin{aligned} & [-1 - (\alpha_{\lambda} - \gamma_{\lambda})] R_{\lambda} F_{\lambda j} - \sum_{\lambda' \neq \lambda} (\beta_{\lambda} - \epsilon_{\lambda\lambda'}) R_{\lambda'} F_{\lambda' j} \\ & + [1 - (\alpha_{\lambda} + \gamma_{\lambda})] R_{\lambda} F_{\lambda j+1} - \sum_{\lambda' \neq \lambda} (\beta_{\lambda} + \epsilon_{\lambda\lambda'}) R_{\lambda'} F_{\lambda' j+1} = R_{\lambda} \frac{(\Delta x \cdot \frac{1}{2} S)}{\mu_{\lambda}} \end{aligned}$$

The parameters $x, S, \int^s$, and $\int^t$ are input for each region, so the variables in parentheses are calculated differently from region to region. Thus, the sets of coefficients of the $F_{\lambda j}$'s are the same for each pair of adjacent points within a certain region.

System of Linear Equations

Based on the above equation, a set of equations is generated for each pair of adjacent points, consisting of an equation for each angle. The number of coefficients in each is twice the number of angles. In addition,

the boundary condition equation, $F(\mu_\lambda) = -F(-\mu_\lambda) = 0$, is applied to the angular fluxes at the first and last points. The number of such equations is equal to the number of angles.

Thus, a system of linear equations is generated, with an equation for each angular flux involved. Within each equation, the order of the flux terms is:

$$F_{1j} \ F_{2j} \ F_{3j} \ \cdots \ F_{NLj} \ F_{1(j+1)} \ F_{2(j+1)} \ F_{3(j+1)} \ \cdots \ F_{NL(j+1)} \ \cdots$$

where NL is the number of angles, and the angles are in decreasing order (see Appendix A). All other coefficients are zero, so the code uses only the necessary storage for the coefficients and the constant term. It may be noted that in the actual code, the equations are generated in the order $\lambda = 1, 2, 3, \dots, NL$, but for each set, the equations for the last λ 's precede those for the first half (positive angles). Ordering the equations in this manner avoided the problem of a zero diagonal element occurring during the simple elimination method of solution for cases with six or more angles.

As can be seen from Figure 1, the matrix system to be solved consists of blocks of matrices along the diagonal, and zeroes elsewhere. This quasi-tridiagonal matrix may therefore be solved by Schechter's method, for which the matrix is broken up into submatrices as indicated in Figure 2. It should be noted that the right half of the E matrices are zero. See Appendix B for a description of the method.

Figure 1. Matrix system for 4 angles and 2 points
in each of 2 regions.

Figure 2. Submatrix designations for Schechter's method of solution.

III. DESCRIPTION OF COMPUTER PROGRAM 790/RE204

November 31, 1962

TITLE: One Group Transport Code for Plane Geometry and Isotropic
Source based on direct partitioned matrix inversion.

METHOD OF SOLUTION: The transport equation (1) for plane geometry and inelastic scattering, is utilized to provide expressions for the first and second derivatives of the angular flux at different points in space.

$$\mu \cdot \frac{\partial F(x, \mu)}{\partial x} + F(x, \mu) \cdot \Sigma^t = \left[\frac{1}{2} S(x) + \frac{\Sigma^s}{2} \int_{-1}^1 F(x, \mu') d\mu' \right]. \quad (1)$$

The derivatives thus obtained, are inserted in second order Quade Hermite integration formulas (2) relating fluxes for up to 16 angles at adjacent points.

$$F_{\lambda j+1} - F_{\lambda j} = \frac{\Delta x}{2} \left(\frac{\partial F_{\lambda j+1}}{\partial x} + \frac{\partial F_{\lambda j}}{\partial x} \right) - \frac{\Delta x^2}{12} \left(\frac{\partial^2 F_{\lambda j+1}}{\partial x^2} - \frac{\partial^2 F_{\lambda j}}{\partial x^2} \right). \quad (2)$$

Combining these with boundary conditions that $F_{\lambda} = F_{-\lambda}$, gives a system of linear equations which is solved for the angular fluxes at up to 600 points in up to 24 regions. The matrix involved consists of zeros and blocks of smaller matrices, so Schechter's system* for solving Quasi-Tridiagonal Matrices is used. (See Appendix B.)

* NYO-2542: Quasi-Tridiagonal Matrices and Type Insensitive Difference Equations by Samuel Schechter of New York University (May 1, 1959).

Finally, the scalar fluxes at various points are calculated (3) using either a Single Gaussian, a Double Gaussian, or a Midpoint Integration formula, for which a coefficient and weight for each angle are given as input. (See Appendix A.)

$$\phi_j = \int_{-1}^1 F_{\lambda',j} d\lambda' = \sum_{\text{all } \lambda} R_{\lambda} F_{\lambda,j} \quad (3)$$

If an error estimation is called for, the original matrix is multiplied by the angular fluxes, and the results are compared with the original right hand side of the linear system.

INPUT DIRECTIONS: Fill out Input Data Form II for each case as follows:

<u>Card Type</u>	<u>Columns</u>	<u>Name</u>	<u>Description</u>
1	1	Carriage Control	= 0 for a double space.
	2-6	NR	Number of regions (up to 24).
	7-12	NL	Number of angles (up to 16, and must be even).
	13-18	ID	<div> -1 To solve system for scalar fluxes but do not estimate errors. 0 No more cases. +1 To calculate errors after solving system. </div>
	19-24	MOUT	<div> -1 Don't output intermediate information. +1 Output matrix and other intermediate information on tape. </div>
	25-64	CASE	Any Hollerith information to identify the output.
2	1-9 10-18 19-27 . . .	INT.C	<p>Fractional Integration Abscissae and Weights, in the order indicated in Appendix A, for the positive angles only. These determine the type of integration formula to calculate scalar fluxes. The values are to be placed to the left in 9 column fields, with no decimal point. Thus, an extra card will only be necessary for more than 8 angles.</p>

INPUT DIRECTIONS: (Cont'd.)

For each region, a card is to contain the following information:

<u>Card Type</u>	<u>Columns</u>	<u>Name</u>	<u>Description</u>
3	1-12	NI	The number of points for which the scalar flux is to be calculated, in addition to the initial point of the region.
	13-24	X	Distance of the far edge of the region from the initial point of the first region (centimeters).
	25-36	S	Source in the region ($\text{cm}^{-3} \text{sec}^{-1}$).
	37-48	SS	Σ^s , the scattering cross section in the region (cm^{-1}).
	49-60	ST	Σ^t , the total cross section in the region (cm^{-1}).

A blank card should be placed after the data for the last case to bring the program to its final stop.

The numbers on card type and also NI are integers and should be placed all the way to the right in the columns specified. The data on card No. 3 may be written with their decimal points anywhere in the columns specified.

FORM OF OUTPUT: The output described below is written on tape to be printed offline. In addition, there is on-line output consisting of the Case Identification, the scalar fluxes at the initial and final points, and the largest deviation (if errors are requested). Sample output is included for a case using a Double Gaussian Integration formula for two different mesh sizes.

DATE AND TIME

CASE ID INFORMATION: NR, NL, ID, MOUT, ..., NE, NP

Number of regions.
 Number of angles.
 ID (If positive, output error estimates).
 MOUT (If positive, output intermediate information).
 Number of equations (= number of angular fluxes to be calculated).
 Number of points at which scalar fluxes are calculated.

INTEGRATION CONSTANTS

For half the angles, the positive abscissae are given in decreasing order, followed by the set of corresponding weights.

(See Appendix A.)

CASE ID INFORMATION:

This is the same as before, but ID or MOUT may have been changed to 0 to indicate something is wrong.

FLUXES:

Angular fluxes ($F_{\lambda j}$) are given, with those for each point starting on a new line. The fluxes are in the same order as the integration constants, with values for decreasing positive angles given first, followed by values for increasing negative angles. (See Appendix A.)

PHI:

The scalar flux (ϕ_j) is given at each point starting with the origin (initial point of the first region) and continuing through the regions, with 12 values on a line.

This concludes the normal output. However, if error estimates are requested, the following is also obtained.

ERROR:

The differences between the original right hand side and the result of multiplying the original matrix by the solution vector. These deviations are output to 2 significant figures for each equation, with 12 values on a line.

RHS:

The right hand sides of each equation are output in the same order as the errors, to provide a basis of comparison. They are printed with 5 places after the decimal point, eight values on a line.

It should be noted that only the precision of the computation may be judged by these criteria, and there is no indication of whether more points or angles should be used for better accuracy.

MACHINE: IBM 704 with 32K memory (FORTRAN II Code)

ACCURACY: The angular fluxes are generally accurate to at least five significant figures.

RUNNING TIME: The time required to compute a case with no error estimation, may be determined as follows:

$$(\text{No. of seconds}) = (\text{No. of angles})^2 \cdot (\text{No. of points}) \div 40.$$

SCRATCH TAPES
NEEDED: 0, 1, and 6 (for printed output)

FINAL STOP: HPR1065₈

APPENDIX A: INTEGRATION CONSTANTS

Abscissae and weights are given for the three types of integrations used in testing the program. Other types of integrations may be used by inputting the appropriate constants.

All of the abscissae take on + and - signs; i.e.: the constants for the Midpoint Integration Formulae for 10 angles are dealt with in the code as follows:

λ	Abscissae	Weights
1	.9	.2
2	.7	.2
3	.5	.2
4	.3	.2
5	.1	.2
6	-.1	.2
7	-.3	.2
8	-.5	.2
9	-.7	.2
10	-.9	.2

n	SINGLE GAUSS		DOUBLE GAUSS		MIDPOINT	
NO. OF ANGLES	ABSCISSAE	WEIGHTS	ABSCISSAE	WEIGHTS	ABSCISSAE	WEIGHTS
2	.57735 02692	1.0	.5	1.0	.5	1.0
4	.86113 63116 .33998 10436	.34785 48451 .65214 51549	.78867 51346 .21132 48655	.5 .5	.75 .25	.5 .5
6	.93246 95142 .66120 93865 .23861 91861	.17132 44924 .36076 15730 .46791 39346	.88729 83346 .5 .11270 16654	.27777 77778 .44444 44444 .27777 77778	.83333 33333 .5 .16666 66667	.33333 33333 .33333 33333 .33333 33333
8	.96028 98565 .79666 64774 .52553 24099 .18343 46425	.10122 85363 .22238 10345 .31370 66459 .36268 37834	.93056 81558 .66999 05218 .33000 94782 .06943 18442	.17392 74226 .32607 25774 .32607 25774 .17392 74226	.875 .625 .375 .125	.25 .25 .25 .25
10	.97390 65285 .86506 33667 .67940 95683 .43339 53941 .14887 43390	.06667 13443 .14945 13492 .21908 63625 .26926 26672 .29552 42247	.95308 99230 .76923 46551 .5 .23076 53449 .04691 00770	.11846 34425 .23931 43352 .28444 44444 .23931 43352 .11846 34425	.9 .7 .5 .3 .1	.2 .2 .2 .2 .2
12	.98156 06342 .90411 72564 .76990 26742 .58731 79543 .36783 14989 .12533 34085	.04717 53364 .10693 93260 .16007 83285 .20316 74267 .23349 25365 .24914 70458	.96623 47571 .83060 46932 .61930 95930 .38069 04069 .16939 53067 .03376 52429	.08566 22462 .18038 07865 .23395 69673 .23395 69673 .18038 07865 .08566 22462	.91666 66667 .75 .58333 33333 .41666 66667 .25 .08333 33333	.16666 66667 .16666 66667 .16666 66667 .16666 66667 .16666 66667 .16666 66667
14	.98628 38087 .92843 48837 .82720 13151 .68729 29048 .51524 86364 .31911 23689 .10805 49487	.03511 94603 .08015 80872 .12151 85707 .15720 31672 .18553 83975 .20519 84637 .21526 38535	.97455 39562 .87076 55928 .70292 25757 .5 .29707 74243 .12923 44072 .02544 60438	.06474 24831 .13985 26957 .19091 50253 .20897 95918 .19091 50253 .13985 26957 .06474 24831	.92857 14286 .78571 42857 .64285 71429 .5 .35714 28571 .21428 57143 .07142 85714	.14285 71429 .14285 71429 .14285 71429 .14285 71429 .14285 71429 .14285 71429 .14285 71429
16	.98940 09350 .94457 50231 .86563 12024 .75540 44084 .61787 62444 .45801 67777 .28160 35508 .09501 25098	.02715 24594 .06225 35239 .09515 85117 .12462 89713 .14959 59888 .16915 65193 .18260 34150 .18945 06105	.98014 49282 .89833 32387 .76276 62049 .59171 73212 .40828 26787 .23723 37950 .10166 67613 .01985 50717	.05061 42681 .11119 05172 .15685 33229 .18134 18917 .18134 18917 .15685 33229 .11119 05172 .05061 42681	.9375 .8125 .6875 .5625 .4375 .3125 .1875 .0625	.125 .125 .125 .125 .125 .125 .125 .125

APPENDIX B: SCHECHTER'S METHOD OF SOLVING QUASI-TRIDIAGONAL MATRICES

The system of linear equations is divided into submatrices as indicated in Figure 2 (page 12). Other submatrices are then determined as follows:

$$\begin{aligned}
 y_1 &= g_1 \\
 A_1 &= M_1 \\
 \left\{ \begin{array}{l} \text{Invert } A_1 \\ C_2 = D_2 A_1^{-1} \\ y_2 = g_2 - C_2 y_1 \\ A_2 = M_2 - C_2 E_1 \end{array} \right. \\
 \left\{ \begin{array}{l} \text{Invert } A_2 \\ C_3 = D_3 A_2^{-1} \\ y_3 = g_3 - C_3 y_2 \\ A_3 = M_3 - C_3 E_2 \end{array} \right. \\
 \left\{ \begin{array}{l} \text{Invert } A_3 \\ C_4 = D_4 A_3^{-1} \\ y_4 = g_4 - C_4 y_3 \\ A_4 = M_4 - C_4 E_3 \end{array} \right. \\
 \vdots
 \end{aligned}$$

This computation is continued until the last A matrix is inverted. Then the angular fluxes are calculated as follows:

$$\begin{aligned}
 v_f &= A_f^{-1} y_f \\
 v_{f-1} &= A_{f-1}^{-1} (y_{f-1} - E_{f-1} \cdot v_f) \\
 v_{f-2} &= A_{f-2}^{-1} (y_{f-2} - E_{f-2} \cdot v_{f-1}) \\
 \vdots
 \end{aligned}$$

where v_f is the final solution vector.

APPENDIX C: FORTRAN Listing

```

C FINAL 790 / RE 204 / C.CHAMOT / OCT. 1962 / (SP CCTPL1 ONLY)FFORSURE
C NOTE (NO. OF ANGLES TIMES NO. OF POINTS MUST NOT EXCEED 600)
C NR =NO. OF REGIONS(UP TO 24)
C NL =NO. OF ANGLES(UP TO 16)
C ID -1 TO SOLVE SYSTEM BUT DONT CALC. ERRORS
C ID 0 IF ARE NO MORE CASES (WILL BE SET =0 IF ERROR OCCURS)
C ID +1 TO CALC. ERRORS AFTER SYSTEM IS SOLVED
C MOUT OUTPUT MATRIX COEFF. IF 1 , AND OUTPUT OTHER INFO.
C MOUT (NORMALLY INPUT AS -1) (WILL BE SET =0 IF ERROR OCCURS)
C NE =NO OF EQUATIONS (UP TO 600)
C NP =NO OF POINTS (UP TO 250)
CNI(IR)=NO. OF POINTS REGION BY REGION
C
  DIMENSION SCALE(1),CTOTAL(1),XO(1), EM(1),EPS(1),CRHS(1)
  2,R(16),U(16),A(16),PHI(250), DIAG(1),CF(1),I12(2)
  3,NI(24),X(24),S(24),SS(24),ST(24)
  4,CFM(600,32),FLUX(600),RHS(600),ECRHS(600)
  5,DA(16,16),DAE(1),Y(600),G(600),V(600)
  EQUIVALENCE (Y,G,RHS),(V,FLUX),(N,NE),(NRP,NR),(NLT3D2,NLPD2)
  GO TO 1002
1000 PAUSE 1000
1002 READ 100 , NR,NL,ID,MOUT
100 FORMAT ( 1H1I5,3I6,30H
213
2///1H )
  CALL NOW(-6)
  SET UP NECESSARY CONSTANTS
  M=2*NL
  MP1=M+1
  MM1=M-1
  NLP1=NL+1
  NLD2=NL/2
  NLD2P1=NLD2+1
  NLPD2=NL+NLD2
  NL1PD2=NLPD2+1
  NLD2PM=NLD2+M
  EM(1)=M
  IF ( ID ) 6020,1060,6020
1060 END FILE 6
  PAUSE 1065
  GO TO 1002
C READ INTEGRATION COEFFICIENTS AND WEIGHTS
6020 READ 602, (U( LL),LL=1,NLD2),(R( LL),LL=1,NLD2)
602 FORMAT (8F9.9)
  DO 6080 LL=1,NLD2
  LLL=NLP1-LL
  U( LLL)=-U( LL)
6080 R( LLL)=R( LL)
C B2 790/RE 204/ MATGEN Y O Y /C.CHAMOT/JUNE 15,1962
  II=NLD2
C READ SOURCE TERMS ETC. FOR EACH REGION
  READ 610 , (NI(IR),X( IR),S( IR),SS( IR),ST( IR),IR=1,NRP)
610 FORMAT (I12,4F12.6)
  NP=1
  DO 6130 IR=1,NRP
  S(IR)=S(IR)/2.0
6130 NP=NP+NI(IR)
  N=NL*NP
C CHECK ON DIMENSIONS
  IF(NP-250)6180,6180,6160
6160 PAUSE 6160
  GO TO 6890
6180 IF (N-600)6190,6190,6160

```

```

S6190 STZ XO(1)
C   CALCULATE ALLES FOR EACH REGION
      DO 6610 IR=1,NRP
      NIR=NI(IR)
      DX  =(X( IR)-XO(1))/(FLOAT(NIR  *2))
      DXSQ=DX  *DX  /3.0
      XO(1)=X( IR)
C   CALCULATE CONSTANTS FOR EACH ANGLE (IN THIS REGION)
      DO 6260 L=1,NL
6260  A( L)=-ST( IR)/(U( L)*R( L))
      DO 6490 L=1,NL
      B  =SS( IR)/(2.0*U( L))
      D  =A( L)*R( L)*(A( L)+2.0*B  )
      ALPHA  =R( L)*DX  *(A( L)+B  )
      BETA   =R( L)*DX  *B
      GAMMA  =-R( L)*DXSQ  *D
      EPSCF  =-R( L)*DXSQ  *B
      NLP1ML=NLP1-L
      II=II+1
      IC=II+NLD2
      ICM1=IC-1
      IF(XMODF(ICM1,NL)-NLD2)6360,6355,6355
6355  IC=II-NLD2
C   CALCULATE CONSTANTS FOR EACH COMBINATION OF L S
6360  DO 6480 LL=1,NL
      LLL=LL+NL
      IF (L-LL)6390,6460,6390
6390  IF (NLP1ML-LL)6420,6400,6420
S6400 STZ EPS(1)
      GO TO 6430
6420  EPS(1)=EPSCF  *(A( L)*R( L)+R( LL)*A( LL))
C   CALCULATE COEFS. FOR L NOT EQUAL TO LL VIA EPS
6430  CFM( IC,LL)=- (BETA  -EPS(1))*R( LL)
      CFM( IC,LLL)=- (BETA  +EPS(1))*R( LL)
      GO TO 6480
C   CALCULATE COEFS. FOR L=LL VIA ALPHA AND GAMMA
6460  CFM(IC,LL)=(-1.0-(ALPHA-GAMMA))*R(LL)
      CFM(IC,LLL)=(1.0-(ALPHA+GAMMA))*R(LL)
6480  CONTINUE
6490  RHS( IC)=S( IR)*R( L)*DX  /U( L)*2.0
C   THE FIRST SET OF COEFFICIENTS FOR THE
C   REGION HAS BEEN CALCULATED
C   TEST IF IS MORE THAN ONE POINT IN REGION
      IF (NIR-1)6510,6610,6530
6510  PAUSE 6510
      GO TO 6890
6530  DO 6600 IN=2,NIR
      DO 6590 L=1,NL
      II=II+1
      I=II-NL
      DO 6580 LL=1,M
6580  CFM( II,LL)=CFM( I,LL)
6590  RHS( II)=RHS( I)
6600  CONTINUE
6610  CONTINUE
C   ZERO OUT THE COEFFICIENTS ETC. FOR BC
      I=0
      DO 6690 IC=1,2
      DO 6680 L=1,NLD2
      I=I+1
      DO 6670 LL=1,M
S6670 STZ CFM(I,LL)

```

```

S6680 STZ RHS(I)
6690 I=II
C   SET UP BOUNDARY CONDITIONS
      I=0
      LO=NL
      LF=MP1
      DO 6840 IC=1,2
      DO 6800 L=1,NLD2
      I=I+1
      LO=LO+1
      LF=LF-1
      CFM( I,LO)=1.0
6800 CFM( I,LF)=-1.0
      I=II
      LO=0
      LF=NLP1
6840 CONTINUE
C   MATRIX IS NOW COMPLETE
      GO TO 7000
C   SET MOUT=0 IF THERE WAS AN ERROR
6890 MOUT=0
C   NRP STILL = NO. OF REGIONS . NP AND NE ARE AVAILABLE.
C   THERE WAS AN ERROR IF MOUT=0 NOW.
C   OUTPUT MATRIX ETC.
7000 PRINT 100 , NR,NL,ID, MOUT,N,NP
      WRITE OUTPUT TAPE 6 , 100 , NR,NL,ID, MOUT,N,NP
      WRITE OUTPUT TAPE 6,603,(U(LL),LL=1,NLD2),(R(LL),LL=1,NLD2)
603  FORMAT (1H0 12F9.5/(1H 12F9.5))
      NN=NLD2
      REWIND 10
      DO 1265 II=1,N
      IF(MOUT)1250,1250,1200
1200 WRITE OUTPUT TAPE 6 , 120 , (CFM( II,I),I=1,M)
120  FORMAT (1H 13F9.5)
      IF (XMODF(II,NN))1220,1240,1250
1220 PAUSE 1220
      GO TO 1000
1240 NN=NL+NN
      WRITE OUTPUT TAPE 6 , 125
125  FORMAT (1H )
1250 IF ( ID ) 1265,1000,1260
1260 WRITE TAPE 10, (RHS( II),(CFM( II,I),I=1,M))
1265 CONTINUE
      END FILE 10
      WRITE OUTPUT TAPE 6 , 100 , NR,NL,ID ,MOUT,N,NP
      IF ( MOUT ) 8100,1000,8100
C   SUBROUTINE CCTPL1 / QUASI MATWIX
C   F2 CCTPL1 / MAY 1962 / NO TAPES
C   FIND NP+1 INVERSES
8100 NPP=1
CMAORD=ORDER OF MATRIX A =ORDER OF MATRIX M (=NL LATER)
CIMEL1=ROW OF STORAGE FOR FIRST ELEMENT OF MATRIX FOR MATINV.
      IMEL1=1
      MAORD=NLD2
CMDORD=NO. OF ROWS IN MATRIX D (ALWAYS NLD2 RELEVANT COLUMNS)
      MDORD=NLD2
CJMEL1=COLUMN OF STORAGE FOR FIRST ELEMENT OF MATRIX FOR MATINV.
      JMEL1=NLP1
C   REPEAT MATRIX INVERSION FOR EACH
8150 CALL MATINV (CFM(IMEL1,JMEL1),MDORD,DET)
C   RESET MDORD REGULAR, LAST, ALL THRU
      IF (NPP-NP) 8190,8500,8550

```



```

8190 MDORD=NL
8200 IMNEW=IMEL1+MAORD
      IMOLD=IMNEW-NLD2
      IMF=IMNEW-1
C      D IS ALWAYS AT (IMNEW,1) AND IS NL*NLD2 UNTIL LAST TIME.
      DO 8275 IDA=1,MDORD
      DO 8275 JDA=1,MAORD
S8275 STZ DA(IDA,JDA)
      IID=IMNEW
      DO 8390 IDA=1,MDORD
        JM=JMEL1
        IY=IMEL1
        DO 8380 JDA=1,MAORD
          JD=1
          DO 8370 IM=IMOLD,IMF
            DA(IDA,JCA)=DA(IDA,JDA)+CFM(IID,JD)*CFM(IM,JM)
8370 JD=JD+1
      Y(IID)=G(IID)-DA(IDA,JDA)*Y(IY)
      IY=IY+1
8380 JM=JM+1
8390 IID=IID+1
C      DA MATRIX HAS MDORD ROWS AND MAORD COLUMNS.
      IM=IMNEW
      DO 8440 IDA=1,MDORD
        JE=NL1PC2
        DO 8430 JM=NLD2P1,NL
          JDA=1
S      STZ DAE(1)
      DO 8410 IE=IMEL1,IMF
        DAE(1)=DAE(1)+DA(IDA,JDA)*CFM(IE,JE)
8410 JDA=JDA+1
      CFM(IM,JM)=CFM(IM,JM)-DAE(1)
8430 JE=JE+1
8440 IM=IM+1
      JMEL1=NLD2P1
      IMEL1=IMNEW
      MAORD=NL
      NPP=NPP+1
      IF(IMNEW-NE)8150,8150,9000
C      READY TO INVERT
8500 MDORD=NLD2
      GO TO 8200
C      ALL A INVERSES ARE IN M POSITIONS OF CFM.
8550 IF ( MOUT-1 ) 8600,8600,8560
8560 WRITE OUTPUT TAPE 6 , 100 , NR,NL,ID, MOUT,N,NP
      DO 8580 II=1,N
8580 WRITE OUTPUT TAPE 6 , 120 , (CFM( II,I),I=1,M)
8600 IDZ=NE-NLD2+1
      IDF=NE
C      RHS NOW CONTAINS Y (=G-C*PREVIOUS Y),SO SOLVE FOR V (=FLUX).
C      IDZ AND IDF ARE STILL AVAILABLE , AS ARE A INVERSES AT M.
8850 JAZ=NLD2P1
      JAF=NL
      IF ( MOUT-1 ) 8860,8860,8855
8855 WRITE OUTPUT TAPE 6 , 115 , (Y(II),II=1,N)
8860 DO 8930 IA=IDZ,IDF
S      STZ V(IA)
      IY=ID7
      DO 8930 JA=JAZ,JAF
        V(IA)=V(IA)+CFM(IA,JA)*Y(IY)
8930 IY=IY+1
C      SET UP Y(F-1)=Y(F-1)-E(F-1)*V(F) ETC. FOR NLD2 VEES ALLES

```

```

      IF(NLD2P1-ICZ)9810,9800,1370
9800 IDZMNL=ICZ-NLD2
      IF(IDZMNL-1)1000,9820,1000
9810 IDZMNL=ICZ-NL
9820 IDZM1=ICZ-1
      DO 9870 IY=IDZMNL, IDZM1
      IV=IDZ
      DO 9870 JE=NLPD2,M
      Y(IY)=Y(IY)-CFM(IY,JE)*V(IV)
9870 IV=IV+1
      IDZ=IDZ-1
      IF(NLD2P1-ICZ)8980,8960,1370
8960 JAZ=NLP1
      IDZ=IDZ+NLD2
8980 IDZ=IDZ-NL
      JAF=NLPD2
      GO TO 8860
9000 ID=0
C      OUTPUT FLUXES
1370 NF=0
      DO 1375 I=1,NP
      NO=NF+1
      NF=NF+NL
1375 WRITE OUTPUT TAPE 6 ,144 ,(FLUX( I),IN=NO,NF)
144  FORMAT (1H F9.6,11F10.6)
C      THERE WAS AN ERROR IF ID=0 NOW
      IF(ID)1390,1000,1390
1390 CONTINUE
C      SOLVE FOR PHIS
      I=0
      DO 1460 IP=1,NP
S      STZ PHI(IP)
      DO 1450 IT=1,NL
      I=I+1
1450 PHI( IP)=PHI( IP)+R( IT)*FLUX( I)
1460 CONTINUE
      PRINT 147 , PHI(1),PHI(NP)
147  FORMAT (6HOPHI =12F9.5/(6X12F9.5))
      WRITE OUTPUT TAPE 6 , 147 , (PHI( I),I=1,NP )
      IF ( ID ) 1002,1000,5724
C      READ MATRIX BACK IN
5724 REWIND 10
      DO 5725 II=1,N
5725 READ TAPE 10 , (RHS( II),(CFM(II,I),I=1,M))
C      CALCULATE DISCREPANCIES , IF ANY (HA)
      IIKEY=NLD2
      IC=1
      JO=NLP1
      JF=M
      IO=0
      DO 5940 II=1,N
S      STZ CRHS(1)
      I=10
      DO 5860 J=JO,JF
      I=I+1
C      TEST IF ARE IN LAST BC
      IF (I-N) 5860,5860,5870
5860 CRHS(1)=CRHS(1)+CFM( II,J)*FLUX( I)
5870 ECRHS( II)=CRHS(1)-RHS( II)
      IF (II-IIKEY) 5940,5900,5890
5890 PAUSE 5777
      GO TO 1000

```

```
5900 GO TO (5920,5910),IC
5910 IO=IO+NL
5920 IIKEY=IIKEY+NL
      IC=2
      JO=1
5940 CONTINUE
      WRITE OUTPUT TAPE 6 , 145 ,(ECRHS( I),I=1,N)
145  FORMAT (6HOERROR 1P12E9.1/(8X12E9.1))
S    STZ ERROR
      DO 5980 I=1,N
5980 ERROR=MAX1F(ERROR,ABSF(ECRHS( I)))
      PRINT 145 , ERROR
      WRITE OUTPUT TAPE 6 , 115 , (RHS( II),II=1,N)
115  FORMAT (6HORHS =8F13.5/(6X8F13.5))
      GO TO 1002
      END(0,1,0,0,0)
```

APPENDIX D: SAMPLE PROBLEM

704 INPUT DATA
FORM II

PROBLEM 790/RE204

ORIGINATOR C. CHAMOT

DATA I.D. SAMPLE

DATE OCT 1962

PAGE 1 OF 1

[illegible]

SAMPLE ONLINE OUTPUT WITH AND WITHOUT ERROR CALC.

2 6 -1 -1 790/RE204/C.CHAMOT/FINAL DG 54 9

PHI = 2.68646 8.26593

2 16 -1 -1 790/RE204/C.CHAMOT/FINAL DG 48 3

PHI = 2.74788 7.86539

2 6 1 -1 790/RE204/C.CHAMOT/FINAL DG 54 9

PHI = 2.68646 8.26593

ERROR 2.6E-07

2 16 1 -1 790/RE204/C.CHAMOT/FINAL DG 48 3

PHI = 2.74788 7.86539

ERROR 6.1E-06

16.03 OCT 12, 1962

(CASE ID) NR NL ID MCUT NE NP
(INFORMATION) 2 6 1 -1 790/RE204/C.CHAMOT/FINAL DG 64 9

"ANGLES" WEIGTHS
(INTEGRATION) .8873 .5 .1127 .27778 .44444 .27778
(CONSTANTS)

(CASE ID) NR NL ID MCUT NE NP
(INFORMATION) 2 6 1 -1 790/RE204/C.CHAMOT/FINAL DG 64 9

(ANGULAR FLUXES)
POINT# "ANGLE" $\theta_1 = .8873$ $\theta_2 = .5$ $\theta_3 = .1127$ $\theta_4 = -.1127$ $\theta_5 = -.5$ $\theta_6 = -.8873$
1 1.937599 1.361488 .719848 .719848 1.361488 1.937399
2 1.745366 1.190789 .708166 .771309 1.592123 2.164866
3 1.587231 1.071123 .726017 .902750 1.894753 2.429597
4 1.462678 0.997558 .775914 1.250777 2.283574 2.732645
5 1.374816 0.972129 .876236 2.277983 2.770492 3.071388
6 2.173123 2.413559 2.882684 3.254705 3.542335 3.692922
7 2.756903 3.316141 3.687579 3.852681 4.001247 4.024602
8 3.664447 3.875577 4.107199 4.181199 4.204131 4.107074
9 3.961814 4.163585 4.265128 4.265128 4.163584 3.951814

(SCALAR FLUXES) PHI POINT# 1 2 3 4 5 6 7 8 9
2.68686 2.73699 2.88669 3.18662 3.77462 5.98135 7.286 8.02429 8.26595

(DEVIATION FROM RIGHT HAND SIDE) ERROR -0. 0. 0. 1.6E-08 4.3E-08 1.5E-08 -3.2E-08 -8.2E-09 -7.9E-09
-2.2E-07 -2.0E-08 -5.6E-09 1.9E-08 6.8E-08 1.8E-08 -3.3E-08 1.6E-8 -9.8E-09
-6.5E-08 1.5E-08 -1.7E-08 9.7E-06 7.4E-08 4.7E-08 -2.2E-08 -5.4E-8 -2.1E-07
-4.0E-08 -9.7E-08 -2.6E-07 1.5E-07 1.4E-07 4.7E-08 -4.1E-08 -5.4E-8 -1.3E-07
-3.4E-08 -7.1E-08 -2.5E-07 3.6E-08 6.0E-08 6.0E-08

(RIGHT HAND SIDE) RHS = 0. 0. 0. -0. -0. -0.
0. -0. -0. -0. -0. 0.
-0. -0. 0. 0. 0. -0.
0. 0. 0. -0.30809 -0.11111 -0.03913
0.30809 -0.30809 -0.11111 -0.03913 0.03913 0.11111
-0.11111 -0.03913 0.03913 0.11111 0.30809 -0.30809
0.03913 0.11111 0.30809 0. 0. 0.

16.03 OCT 12, 1962

APPENDIX E: Tables to Compare Speed and Accuracy
of Various Methods of Solution

The following tables are based on a case with two regions (uranium and water), with fluxes evaluated at N_1 points in the first region and N_2 points in the second region. Some of the same information has been arranged differently from table to table to facilitate making comparisons between the following:

- i) first and second equations are basis for iteration procedures.
- ii) three types of integration formulas for scalar flux calculation.
- iii) first and second order approximations.
- iv) from two to sixteen angles.
- v) various numbers of points in each region.

It should be noted that final values of scalar fluxes are the same for iteration procedures and for matrix methods.

Table 1. Stability of iteration calculations.

Table 2. Number of iterations necessary to obtain five figure accuracy (i.e., less than .00001 change in ϕ_1).

Table 3. Number of iterations necessary to reach various accuracies, arranged to study 1) effect of varying number of angles and 2) effect of varying number of points.

Table 4. Number of iterations necessary to reach various accuracies, arranged to compare single Gauss and double Gauss for first equation only.

Table 5. Number of iterations necessary to reach various accuracies for midpoint integration for first equation only.

Table 6. Number of iterations necessary to reach various accuracies, arranged to compare single Gauss and midpoint for second equation only.

Table 7. Machine time for a representative case run with various iteration schemes; to compare the usual time involved in calculating a case, with that for starting with a rough mesh and increasing the number of points, or the number of angles.

- Table 8. Calculated values of ϕ_1 , the scalar flux of the initial point of the first region (uranium), for various mesh sizes, arranged to study changes with increased number of points and angles.
- Table 9. Calculated values of ϕ_2 , the scalar flux at the boundary point between regions, for various mesh sizes, arranged to study changes with increased number of points and angles.
- Table 10. Calculated values of ϕ_3 , the scalar flux at the final point of the second region (water), for various mesh sizes, arranged to study changes with increased number of points and angles.
- Table 11. Scalar fluxes (ϕ_1, ϕ_2, ϕ_3) obtained by different methods with four points in each region. The values for the second degree calculation are the same as with more points, but those for the first degree calculation may be improved considerably by increasing the number of points.
- Table 12. Results for various ratios of Σ^s to Σ^t for a representative case, run with the first equation, double Gauss integration, and second order approximation.
- Table 13. Number of iterations required for various initial flux values for a representative case run with the first equation and midpoint integration.
- Table 14. Accuracy obtainable with the matrix methods using single and double precision.
- Table 15. Results obtained for one, two, and four points in each region, using second degree approximation.

*This sample problem had been used in previous studies (E. H. Bareiss: Flexible Transport Theory Routines for Nuclear Reactor Design; Navy Dept. David Taylor Model Basin, Report 1030, (1956)).

TABLE 1. Stability of Iteration Calculations

1 or 2 means cases ran successfully [1 = Equation 1; 2 = Equation 2].

(1) or (2) means cases ran but took a large number of iterations.

① or ② means cases were attempted but were unstable.

$\epsilon=0$ (First degree)

$N_1 = 1 \quad 2 \quad 4 \quad 4 \quad 4 \quad 16$

$N_2 = 1 \quad 2 \quad 4 \quad 8 \quad 16 \quad 16$

No. of angles	SINGLE GAUSS				
2	1	1	1	1	1
4	1				
6	1				
8	1				
10	1				
12	1				
14	1				
16	1	1	1		

No. of angles	DOUBLE GAUSS			
2	1			
4	1			
6	1			
8	1			
10	1			
12	1			
14	1			
16	1	1	1	

No. of angles	MIDPOINT GAUSS			
2				
4	1	1		
6		1		
8		1		
10		1		
12		1		
14		1		
16		1	1	1

$\epsilon=1$ (Second degree)

$N_1 = 1 \quad 2 \quad 4 \quad 4 \quad 4 \quad 16 \quad 8$

$N_2 = 1 \quad 2 \quad 4 \quad 8 \quad 16 \quad 16 \quad 24$

No. of angles	SINGLE GAUSS					
2	1	1				
4	1	1,2	1,2	1,②	1	
6	1	1,2				
8	1	1,2				
10	①	1,2				
12	①	1,2				
14	①	1,2				
16	①	(1),2	2	2		

No. of angles	DOUBLE GAUSS					
2	1,2					1
4	1,2	2	②			1
6	1,2					1
8	①,2					1
10	①,2					1
12	①,2					
14	①,2					1
16	①,2	①,2	①,②	①	(1)	

No. of angles	MIDPOINT GAUSS					
2	1	1				1
4	①,2	1,2	1,2	2	②	1,②
6	(1)	1,2	1	1		1
8	①	1,2				1
10	①	1,2				1
12	①	①,2				
14	①	①,2				
16	①	①,2	1,2	1,2		

TABLE 2. Number of Iterations Necessary to Obtain Five Figure Accuracy (i.e., less than .00001 change in ϕ_1)

No. of Angles	No. of Points		1st Equation						2nd Equation					
	N ₁	N ₂	Single Gauss		Double Gauss		Midpoint		Single Gauss		Double Gauss		Midpoint	
			1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.
2	2	2		105				119						
	4	4		92				104						
	8	8						88						
4	1	1					100						110	188
		2					100	100						155
		4					96	96						181
	2	2		113				115					80	129
		4												120
		8				93	91	93						176
	4	4	97	98			98	98		77				88
		8						95						100
		16		90		87	87	90					150	170
	8	8		104		87	92	90						
		32		98	89	85	85	82					155	165
	16	16												
6		64	83	83			84	82					160	165
	32	32						80						
		128					84						150	165
	64	128												
5	2	2												
	4	4						97						
8	2	2												
	4	4												
10	2	2												
	4	4												
12	2	2												
	4	4												
14	2	2												
	4	4												
16	2	2												
	4	4												
		8						144						
		16												
	8	24				300								

TABLE 3. Number of Iterations Necessary to Reach Various Accuracies

No. of Angles	N ₁	N ₂	Accuracy, %	1st Equation						2nd Equation			
				Single Gauss		Double Gauss		Midpoint		Single Gauss		Midpoint	
				1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.
Effect of Varying Angles:													
2	4	4	.5 1 .1 .01 .001		23 37 55 74 92				26 41 62 83 104				
4	4	4	5 1 .1 .01 .001	25 39 59 78 97	25 39 59 78 98			26 40 60 80 98	25 39 59 79 98		20 31 47 62 77		23 35 53 71 88
6	4	4	5 1 .1 .01 .001						25 39 59 78 97				
Effect of Varying Points:													
4	1	4	5 1 .1 .01 .001					28 42 62 81 96	27 41 61 81 96				52 79 118 153 181
	2	8	5 1 .1 .01 .001				25 37 58 73 93	25 38 56 75 91	24 37 56 74 93				46 72 106 140 176
	4	16	5 1 .1 .01 .001		24 38 52 70 90		23 37 52 70 87	23 34 53 69 87	23 35 53 70 90			43 64 97 127 150	43 66 98 133 170
	8	32	5 1 .1 .01 .001		23 36 50 70 98	23 34 49 69 89	23 33 51 69 85	23 34 51 69 85	22 34 51 68 82			42 63 95 126 155	42 65 96 130 165

TABLE 4. Number of Iterations Necessary To Reach Various Accuracies

No. of angles	First Equation: Single Gauss/Double Gauss Integration											
	No. of points		5% accuracy		1% accuracy		.1% accuracy		.01% accuracy		.001% accuracy	
	N ₁	N ₂	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.
2	2	2		<u>27</u>		<u>42</u>		<u>64</u>		<u>85</u>		<u>105</u>
	4	4		<u>23</u>		<u>37</u>		<u>55</u>		<u>74</u>		<u>92</u>
	16	16										
4	1	1										
		2										
		4										
	2	2		<u>29</u>		<u>45</u>		<u>68</u>		<u>91</u>		<u>113</u>
		4										
		8		/25		/37		/58		/73		/93
	4	4	<u>25</u>	<u>25</u>	<u>39</u>	<u>39</u>	<u>59</u>	<u>59</u>	<u>78</u>	<u>78</u>	<u>97</u>	<u>98</u>
		8										
		16		<u>24</u> /23		<u>38</u> /37		<u>52</u> /52		<u>70</u> /70		<u>90</u> /87
	8	8		<u>26</u> /26		<u>37</u> /37		<u>56</u> /56		<u>75</u> /75		<u>104</u> /92
		32	/23	<u>23</u> /23	/34	<u>36</u> /33	/49	<u>50</u> /51	/69	<u>70</u> /69	/89	<u>98</u> /85
	16	16										
	64		<u>22</u>	<u>23</u>	<u>33</u>	<u>34</u>	<u>49</u>	<u>50</u>	<u>68</u>	<u>68</u>	<u>83</u>	<u>83</u>

Note: Underlined values are for Single Gauss Integration.

TABLE 5. Number of Iterations Necessary to Reach Various Accuracies
(1st equation: midpoint integration)

No. of Angles	No. of Points		5% Accuracy		1% Accuracy		.1% Accuracy		.01% Accuracy		.001% Accuracy	
	N ₁	N ₂	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.
2	2	2		30		47		71		95		119
	4	4		26		41		62		83		104
	16	16		23		35		53		71		88
4	1	1	39		59		85		96		100	
		2	30	30	48	47	70	70	90	90	100	100
		4	28	27	42	41	62	61	81	81	96	96
	2	2		29		46		69		92		115
		4										
		8	25	24	38	37	56	56	75	74	91	93
	4	4	26	25	40	39	60	59	80	79	98	98
		8		24		37		55		74		95
		16	23	23	34	35	53	53	69	70	87	90
	8	8		23		38		59		75		90
		32	23	22	34	34	51	51	69	68	85	82
	16	16										
		64	23	22	33	34	50	50	67	66	84	82
	32	32										
		128	22	22	34	34	50	50	(63)	66	(69)	80
	64	128	22		32		52		70		84	
6	2	2										
	4	4		25		39		59		78		97

TABLE 6. Number of Iterations Necessary to Reach Various Accuracies
(2nd equation: single Gauss/midpoint integration)

No. of Angles	No. of Points		5% Accuracy		1% Accuracy		.1% Accuracy		.01% Accuracy		.001% Accuracy	
	N ₁	N ₂	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.
4	1	1	32	49	48	74	71	111	91	148	110	188
		2		41		62		96		123		155
		4		52		79		118		153		181
	2	2	21	34	33	52	49	78	68	103	80	129
		4		32		47		72		96		120
		8		46		72		106		140		176
	4	4	<u>20</u>	23	<u>31</u>	35	<u>47</u>	53	<u>62</u>	71	<u>77</u>	88
		8		27		40		60		80		100
		16		43		66		98		133		170
	8	8	42	42	63	65	95	96	126	130	155	165
		32		42		63		96		129		165
	16	16	42	42	63	63	94	96	126	129	160	165
		64		42		62		94		128		165
	32	32	41	42	62	62	94	94	125	128	150	165
		128		42		62		94		128		165
	64	128										

Note: Underlined values are for Single Gauss Integration.

TABLE 7. Machine Time for a Representative Case Run with Various Iteration Schemes.

- I. The usual method is to start with zero fluxes and iterate to an accuracy of .00001.
- II. Iterate with a rough mesh to an accuracy of .00001, and then with finer meshes, using calculated and interpolated values for fluxes.
- III. Iterate with a rough mesh to a limited accuracy, and then with finer meshes to a better and better accuracy.

Increase number of points for: First Equation, Single Gauss, Second Degree Case										
4 angles		METHOD II		4 angles		METHOD II		METHOD III		
N_1	N_2	No. of Iterations	Mach. Time	N_1	N_2	No. of Iterations	Mach. Time	No. of Iterations	Accuracy	Mach. Time
2	2	134	27 secs.							
4	4	81	32 secs.	4	4	117	47 secs.	38	.05	15 secs.
8	8	65	52 secs.	4	8	65	39 secs.	20	.005	12 secs.
16	16	53	85 secs.	4	16	52	50 secs.	52	.00001	52 secs.
Total Machine Time		195 secs.				136 secs.				79 secs.
Ratio to Method I		1.22				1.36				.79

Increase number of angles for: First Equation, Single Gauss, Second Degree Case								
$N_1=4=N_2$	METHOD II		METHOD III			METHOD III		
No. of Angles	No. of Iterations	Mach. Time	No. of Iterations	Accuracy	Mach. Time	No. of Iterations	Accuracy	Time
2			24	.2	5 secs.	24	.2	5 secs.
4	117	45 secs.	14	.1	6 secs.	14	.1	6 secs.
6	75	45 secs.	19	.01	11 secs.	5	.05	3 secs.
8	72	53 secs.	33	.001	26 secs.	5	.005	4 secs.
10	48	48 secs.	38	.0001	38 secs.	5	.0005	5 secs.
12	60	72 secs.	66	.00001	79 secs.	73	.00001	87 secs.
Total Machine Time		268 secs.				165 secs.		110 secs.
Ratio to Method I		2.24				1.38		.92

TABLE 8. Calculated Values of ϕ_1 for Various Mesh Sizes and Methods of Calculation

No. of Angles	No. of Points		Single Gauss		Double Gauss		Midpoint	
	N ₁	N ₂	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.
2	2	2		¹ 5.5911				
	4	4		¹ 5.5910	5.3119	¹ 5.3291 ²		¹ 5.3291
		8		¹ 5.5909				¹ 5.3291
4	1	1					¹ 4.9161	¹ 05.3290 ²
		2					¹ 4.6906	¹ 5.3187 ²
		4					¹ 4.6686	¹ 5.3189 ²
	2	2		¹ 5.3418				¹ 5.2915 ²
		4						5.2915 ²
		8				¹ 5.2924	¹ 5.1668	¹ 5.2915 ²
	4	4	¹ 5.3184	¹ 5.3411 ²	5.2621	¹ 5.2902 ²	¹ 5.2643	¹ 5.2901 ²
		8		¹ 5.3411 ²		5.2901 ²		¹ 5.2901 ²
		16		¹ 5.3411 ⁰²		¹ 5.2901 ²⁰	¹ 5.2603	¹ 5.2901 ⁰²
	8	8		¹ 5.3411 ²		¹ 5.2900 ²		¹ 5.2900 ²
		32		¹ 5.3411	¹ 5.2821	¹ 5.2900	¹ 5.2826	¹ 5.2900 ⁰²
	16	16		¹ 5.3411				¹ 5.2900 ⁰²
		64	¹ 5.3394	¹ 5.3411			¹ 5.2882	¹ 5.2900 ⁰²
	32	32						⁰²
		128					¹ 5.2892	¹ 5.2900 ⁰²
	64	128					¹ 5.2899	
6	2	2		¹ 5.3189				¹ 5.33
	4	4	¹ 5.2917	¹ 5.3173 ²	¹ 5.3546	¹ 5.3729 ²	¹ 5.3011	¹ 5.3252 ²
8	2	2		¹ 5.3319				¹ 0
	4	4	¹ 5.3045	¹ 5.3294 ²	¹ 5.3596	5.3761 ²	¹ 5.3266	¹ 5.3475 ²
10	2	2		¹ 0				¹ 0
	4	4	¹ 5.3197	¹ 5.3426 ²	¹ 5.3519	5.3693 ²	¹ 5.3390	¹ 5.3580 ²
12	2	2		¹ 0				¹ 0
	4	4	¹ 5.3323	¹ 5.3533 ²	¹ 5.3545	5.3680 ²	¹ 5.3442	¹ 05.3625 ²
14	2	2		¹ 0				¹ 0
	4	4	¹ 5.3399	¹ 5.3594 ²	¹ 5.3609	5.3687 ²	¹ 5.3462	¹ 05.3646 ²
16	2	2		¹ 0				¹ 0
	4	4	¹ 5.3442	¹ 5.3630 ²	¹ 5.3651	5.3694 ²	¹ 5.3499	¹ 05.3680 ²
		8	¹ 5.3406	5.3630 ²	¹ 5.3699		¹ 5.3453	¹ 5.3680 ²
		16	¹ 5.3398	5.3630 ²	¹ 5.3698		¹ 5.3445	¹ 5.3680 ²
	8	24				¹ 5.3687 ²		

Notes: 1 refers to 1st equation; 2 refers to 2nd equation; 0 indicates instability.

TABLE 9. Calculated Values of ϕ_2 for Various Mesh Sizes and Methods of Calculation

No. of Angles	No. of Points		Single Gauss		Double Gauss		Midpoint	
	N ₁	N ₂	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.
2	2	2		7.1039				
	4	4		7.1038	7.2708	7.2794		7.2794
	8	8		7.1037				7.2794
4	1	1					7.1779	7.4707
		2					7.3218	7.5113
		4					7.3167	7.5116
	2	2		7.4083			7.4644	7.5056
		4						7.5056
		8				7.5854	7.4544	7.5056
	4	4	7.3995	7.4080	7.5790	7.5851	7.4982	7.5053
		8		7.4080		7.5851		7.5053
		16		7.4080		7.5851	7.4925	7.5053
	8	8		7.4080		7.5851		7.5053
		24				7.5851		
		32		7.4080	7.5822	7.5851	7.5021	7.5053
	16	16		7.4080				7.5053
		64	7.4071	7.4080			7.5045	7.5053
	32	32						
		128					7.5047	7.5053
	64	128					7.5052	
6	2	2		7.4992				
	4	4	7.4919	7.4990	7.5411	7.5492	7.5281	7.5348
8	2	2		7.5251				
	4	4	7.5187	7.5253	7.5229	7.5427	7.5321	7.5396
10	2	2						
	4	4	7.5259	7.5327	7.5093	7.5437	7.5316	7.5412
12	2	2						
	4	4	7.5296	7.5371	7.4992	7.5424	7.5293	7.5420
14	2	2						
	4	4	7.5303	7.5390	7.4844	7.5395	7.5262	7.5427
16	2	2						
	4	4	7.5298	7.5401	7.4933	7.5363	7.5211	7.5461
		8	7.5289	7.5401	7.5311		7.5351	7.5463
		16	7.5278	7.5401	7.5333		7.5339	7.5463
	16	16						
	8	24				7.5441		

TABLE 10. Calculated Values of ϕ_3 for Various Mesh Sizes and Methods of Calculation

No. of Angles	No. of Points		Single Gauss		Double Gauss		Midpoint	
	N ₁	N ₂	1st deg.	2nd deg.	1st deg.	2nd deg.	1st deg.	2nd deg.
2	2	2		14.952				
	4	4		14.952	17.500	17.484		17.484
	8	8		14.952				17.484
4	1	1					18.599	16.231
		2					16.370	16.775
		4					16.629	16.793
	2	2		16.363			16.504	16.773
		4						16.790
		8				16.584	16.749	16.790
	4	4	16.385	16.372	16.592	16.584	16.800	16.790
		8		16.372		16.584		16.790
		16		16.372		16.584	16.780	16.790
	8	8		16.372		16.584		16.790
		24				16.584		
		32		16.372	16.582	16.584	16.787	16.790
	16	16		16.372				16.790
		64	16.372	16.372			16.789	16.790
	32	32						
		128					16.789	16.790
	64	128					16.790	
6	2	2		16.457				
	4	4	16.490	16.478	16.475	16.532	16.649	16.649
8	2	2		16.455				
	4	4	16.512	16.505	16.393	16.525	16.565	16.597
10	2	2						
	4	4	16.502	16.513	16.367	16.510	16.511	16.573
12	2	2						
	4	4	16.489	16.520	16.362	16.492	16.475	16.559
14	2	2						
	4	4	16.472	16.523	16.361	16.479	16.149	16.549
16	2	2						
	4	4	16.456	16.524	16.362	16.472	16.417	16.544
		8	16.520		16.489		16.548	16.556
		16	16.516	16.526	16.522		16.546	16.556
	16	16						
	8	24				16.532		

TABLE 11. Scalar Fluxes Obtainable from Different Methods with Four Points in Each Region

No. of Angles	X	2nd Degree			1st Degree		
		Single Gauss	Double Gauss	Midpoint	Single Gauss	Double Gauss	Midpoint
2	0	5.5909	5.3291	5.3291		5.3119	
	.8	7.1038	7.2794	7.2794		7.2708	
	1.8	14.952	17.484	17.484		17.500	
4	0	5.3411	5.2901	5.2901	5.3184	5.2621	5.2643
	.8	7.4080	7.5851	7.5053	7.3995	7.5790	7.4982
	1.8	16.372	16.584	16.790	16.385	16.592	16.800
6	0	5.3173	5.3729	5.3252	5.2917	5.3546	5.3011
	.8	7.4990	7.5493	7.5348	7.4919	7.5411	7.5282
	1.8	16.478	16.5319	16.649	16.490	16.475	16.649
8	0	5.3294		5.3475	5.3045	5.3596	5.3266
	.8	7.5253		7.5396	7.5187	7.5229	7.5321
	1.8	16.505		16.597	16.512	16.393	16.565
10	0	5.3426		5.3580	5.3197	5.3519	5.3390
	.8	7.5327		7.5412	7.5250	7.5093	7.5316
	1.8	16.513		16.573	16.502	16.367	16.511
12	0	5.3533		5.3625	5.3323	5.3545	5.3442
	.8	7.5371		7.5420	7.5296	7.4992	7.5293
	1.8	16.520		16.559	16.489	16.362	16.475
14	0	5.3594		5.3646	5.3399	5.3609	5.3462
	.8	7.5390		7.5427	7.5303	7.4944	7.5262
	1.8	16.523		16.549	16.472	16.361	16.449
16	0	5.3630	5.3687	5.3680	5.3442	5.3651	5.3499
	.8	7.5401	7.5441	7.5461	7.5298	7.4933	7.5211
	1.8	16.524	16.532	16.544	16.456	16.362	16.417

Note: Results obtained using the 1st equation were the same as were obtained using the 2nd equation.

TABLE 12. Results for Various Ratios of Σ^s to Σ^t for a Case Run with the First Equation, Double Gauss, and Second Degree Approximation.

$\Sigma^t=3.03$	4 angles; $N_1 = 8 = N_2$			
Σ^s	No. of Iterations	ϕ_1	ϕ_2	ϕ_3
3.01	92	5.2900	7.5851	16.584
0	30	.1735	.3666	.6545
$1/4 \Sigma^t$	30	.2473	.4796	.8665
$1/2 \Sigma^t$	20	.3914	.6927	1.2786
$3/4 \Sigma^t$	27	.7797	1.2500	2.4130
Σ^t	$\oplus$			
$5/4 \Sigma^t$	$\oplus$			

Note: $\oplus$ indicates instability

TABLE 13. Number of Iterations Required for Various Initial Flux Values for a Representation Case Using the First Equation and Midpoint Integration.

Degree	No. of angles	N ₁	N ₂	Number of Iterations	
				F _{initial} = 0	F _{initial} ≠ 0
2	4	8	32	101	116*
2	4	32	128	92	101 [†]
1	4	32	128	70	101 [†]

* F_{initial} = 5

† F_{initial} = 9

TABLE 14. Accuracy Obtainable with Matrix Methods

No. of Angles	No. of Points		No. of Equations	Single Precision						Double Precision	
	N ₁	N ₂		Simple Elimination			Schechter's Block Method			Simple Elimination	Schechter's Block Method
				Single Gauss	Double Gauss	Midpoint	Single Gauss	Double Gauss	Midpoint	Single Gauss	Single Gauss
2	1	1	6	10 ⁻⁵	10 ⁻⁷	10 ⁻⁷		✓			
	64	64	258				10 ⁻⁷				
4	1	1	12	10 ⁻⁵	10 ⁻⁷	10 ⁻⁷	10 ⁻⁶	✓	10 ⁻⁶	10 ⁻¹⁴	10 ⁻¹⁴
	16	32	196	10 ⁻⁵	10 ⁻⁷	10 ⁻⁷					
6	1	1	18	10 ⁻⁵	10 ⁻⁶	10 ⁻⁷		✓			
	16	32	294				10 ⁻⁶				
	4	4	54					10 ⁻⁶			
8	1	1	24	10 ⁻⁵	10 ⁻⁵	10 ⁻⁵		✓			
10	1	1	30	10 ⁻⁵	10 ⁻⁵	10 ⁻⁵		✓			
	8	3	170	10 ⁻⁵	10 ⁻⁴	10 ⁻⁴					
12	1	1	36	10 ⁻²	10 ⁻²	10 ⁻³		✓			
	4	4	108				10 ⁻⁷				
	8	16	300				10 ⁻⁶				
14	1	1	42	10 ⁻²	10 ⁻¹	10 ⁻²		✓			
16	1	1	48	10 ⁻¹	22	10 ⁻¹	10 ⁻⁶	10 ⁻⁵		10 ⁻⁷	10 ⁻¹⁴
	4	4	144	10 ⁻²	10 ⁻¹	10 ⁻¹					

Note: The check marks indicate the case ran satisfactorily, but errors were not calculated.

TABLE 15. Results Obtained for One, Two, and Four Points in Each Region,
Using Second Degree Approximations

	ϕ_1			ϕ_2			ϕ_3		
No. of Points:	1:1	2:2	4:4	1:1	2:2	4:4	1:1	2:2	4:4
No. of Angles.									
Double Gauss									
2	5.3331	5.3291	5.3291	7.2020	7.2796	7.2794	17.4848	17.4841	17.484
4	5.3572	5.2925	5.2902	7.5165	7.5853	7.5851	15.7861	16.5495	16.584
6	5.5087	5.3785	5.3729	7.3833	7.5416	7.5492	15.7037	16.3667	16.532
8	5.5147	5.3846	5.3761	7.3739	7.5133	7.5427	15.7182	16.2944	16.525
10	5.5000	5.3839	5.3693	7.3846	7.4967	7.5437	15.7275	16.2807	16.510
12	5.4953	5.3898	5.3680	7.3879	7.4884	7.5424	15.7312	16.2796	16.492
14	5.4952	5.3948	5.3687	7.3878	7.4871	7.5395	15.7312	16.2806	16.479
16	5.4958	5.3965	5.3694	7.3875	7.4879	7.5363	15.7308	16.2821	16.472
Single Gauss									
2	5.5931	5.5911	5.5910	7.1054	7.1039	7.1038	14.9532	14.952	14.952
4	5.3557	5.3418	5.3411	7.4028	7.4083	7.4080	16.2084	16.363	16.372
6	5.3603	5.3189	5.3173	7.4585	7.4992	7.4990	15.8943	16.457	16.478
8	5.4014	5.3319	5.3294	7.4457	7.5251	7.5253	15.8208	16.455	16.505
10	5.4356		5.3426	7.4247		7.5327	15.7802		16.513
12		5.3575	5.3533		7.5329	7.5370		16.4048	16.5197
14			5.3594			7.5390			16.523
16	5.4334	5.3684	5.3630	7.3956	7.5287	7.5401	15.7411	16.3651	16.5242
Midpoint									
2			5.3291			7.2794			17.484
4	5.3290	5.2915	5.2901	7.4707	7.5056	7.5053	16.231	16.773	16.790
6			5.3252			7.5348			16.649
8			5.3475			7.5396			16.597
10			5.3580			7.5412			16.573
12			5.3625			7.5420			16.559
14			5.3646			7.5427			16.549
16			5.3680			7.5461			16.544

Note: Results obtained using matrix methods were the same as were obtained using an iteration procedure.