72
4-18
LA-4600

2113

MASTER

Theory and Use of the Spherical Harmonics,
First Collision Source, and
Variable Weight Versions of the
TWOTRAN Transport Program

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7400-ENG. 28

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

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.

Printed in the United States of America. Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161
Price: Printed Copy \$3.00; Microfiche \$0.95

LA-4600
UC-32, UC-80

ISSUED: March 1972

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

Theory and Use of the Spherical Harmonics, First Collision Source, and Variable Weight Versions of the TWOTRAN Transport Program

by

K. D. Lathrop
F. W. Brinkley
P. Rood

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.

RESTRICTION OF THIS DOCUMENT IS UNLIMITED

I.	INTRODUCTION	1
II.	VARIABLE WEIGHTED DIAMOND SPATIAL DIFFERENCING - TWOTRAN VM	2
III.	CONVERSION OF DISCRETE ORDINATES EQUATIONS TO EQUATIONS LIKE SPHERICAL HARMONICS EQUATIONS - TWOTRAN PM	3
A.	Theory of the Conventional Truncation Source - TWOTRAN PM	4
B.	Theory of an Alternate Truncation Source	7
C.	General Remarks on Truncation Sources	7
D.	Typical Results Obtained with TWOTRAN PM	7
E.	Coding Changes to Convert TWOTRAN to TWOTRAN PM	9
IV.	ADDITION OF A FIRST COLLISION SOURCE IN CYLINDRICAL GEOMETRY - TWOTRAN PC	13
A.	General Theory	13
B.	Code Changes	14
C.	Typical Results	16
V.	TWOTRAN STRUCTURE CHANGES	17
A.	Overlay and Common Changes	17
B.	Flux Dumps and Problem Restart	23
C.	Additions to the Restart Tape	23
1.	TWOTRAN PC	23
2.	TWOTRAN PM	24
D.	Additional Changes to TWOTRAN	24
	REFERENCES	25

THEORY AND USE OF THE SPHERICAL HARMONICS, FIRST COLLISION SOURCE,
AND VARIABLE WEIGHT VERSIONS OF THE TWOTRAN TRANSPORT PROGRAM

by

K. D. Lathrop, F. W. Brinkley, and P. Rood

ABSTRACT

Described are three new versions of the TWOTRAN program for solution of the multigroup discrete ordinates approximation to the two-dimensional Boltzmann transport equation. The first of these versions, TWOTRAN FH, is designed to solve discrete ordinates equations that have been converted to equations like spherical harmonics equations to eliminate ray effects. The second version, TWOTRAN FC, computes the analytic flux due to a point source on the axis of a finite cylinder and automatically uses this flux to compute the first collision source of particles for further transport. The third version, TWOTRAN VW, uses a variable weight diamond difference scheme for calculating spatial derivatives. All three versions operate from the basic TWOTRAN program and offer the options available in that program for three two-dimensional geometries. Typical results of applying each of the versions of TWOTRAN are given.

I. INTRODUCTION

Once a major two-dimensional transport code has been made operational and has been thoroughly checked for error, it is usually more profitable to develop new methods by altering the existing code rather than by developing a new program. This is true since file manipulation routines can selectively alter a program without destroying the original. We have developed three special versions, of varying complexity, for the TWOTRAN program¹ by using the Los Alamos Scientific Laboratory (LASL) Update² system. These versions, called TWOTRAN FH, for spherical harmonics, TWOTRAN FC, for first collision source, and TWOTRAN VW, for variable weight, all use the major part of the TWOTRAN package without change. This report records the theory of these versions and the mechanics of the changes made to TWOTRAN. We also provide input instructions for the small amount of additional input required by these programs. For example, in TWOTRAN FH, the user

must indicate the order of the spherical harmonics (P_N) approximation he wishes to use.

The three versions are described in separate, self-contained sections of this report. Changes made to the TWOTRAN program are described in enough detail for a user to understand the reason for the change, and alterations in storage requirements are listed in detail. The notation used throughout is consistent with that of the TWOTRAN manual.

Since the user's "Manual" for TWOTRAN was issued, there have been a number of changes besides the ones described in this report. These changes include:

1. Placing much of the in-line coding in sub-routine INNER into two subroutines, IN and OUT, for calculating the inward and outward directed fluxes. Although these routines are called twice for each vertical cell per inner iteration, a net decrease of about 10% in running time is obtained on CDC machines

because dimensionality of several variables is reduced in the subroutines.

2. A pointwise density variation of the form $P(x)Q(y)$ is allowed for the cross sections used in the calculation. This change had been made in the program as released to the national code centers.

3. Cross sections in the FIDO format (ANISE loader) can now be read as input.

4. Specification of a value of G for a parametric eigenvalue is now allowed.

5. Output printing can be suppressed for all or portions of the input flux guess, final flux, cross sections, final fission source, normalized source, or the coarse-mesh balance tables.

6. If the coarse-mesh balance table printing is suppressed, one fewer outer iteration is performed. Heretofore, the only purpose of the last outer iteration was to create three tables.

7. Core size requirement adjustment routines, mode error dump routines, and the MAXCON subroutine have been removed. The latter was replaced by in-line calls to ECHM or ECHD, two simple GEC routines for using Extended Core.

8. The overlay and dump formats were changed.

9. A general cleanup of equivalences and named common blocks was performed.

10. The program was modified to allow disk storage as required by problems using more storage than available in ECS or LCM. Use of special I/O routines allow such problems to be executed with only a 10% increase in running time on the CDC-7600.

For completeness, the changed structure of TWO-TAN is sketched in Sec. V. Because the changes of item 10, above, are peculiar to the LANS computing environment, they are not described. Throughout we use the symbols ECS (Extended Core Storage) and LCM (Large Core Memory) interchangeably.

II. VARIABLE WEIGHTED DIAMOND SPATIAL DIFFERENCING - TWO-TAN VW

The original TWO-TAN uses the Diamond difference assumptions³ that equate the sum of cell edge fluxes to twice the cell-centered flux. This scheme is more accurate than any other explicit scheme known, but is not positive. Consequently, a negative flux fixup is used to insure positive fluxes. In TWO-TAN VW, the positive, variable

weighted diamond scheme⁴ is used for the spatial variables and the negative flux fixup is eliminated. The variable weighted scheme gives smoother fluxes and interacts more favorably with convergence acceleration procedures than the diamond scheme, but is usually less accurate.

All of the changes necessary for TWO-TAN VW are made in subroutines IN and OUT. The equation to be solved is [Eq. (51) of Ref. 1]

$$\begin{aligned} & u(A_{1+\frac{1}{2}}N_{1+\frac{1}{2}} - A_{1-\frac{1}{2}}N_{1-\frac{1}{2}}) \\ & + (A_{1+\frac{1}{2}} - A_{1-\frac{1}{2}})(G_{2+\frac{1}{2}}N_{2+\frac{1}{2}} - G_{2-\frac{1}{2}}N_{2-\frac{1}{2}})/N_m \\ & + v(N_{j+\frac{1}{2}} - N_{j-\frac{1}{2}}) + c_e vE = vS. \end{aligned} \quad (1)$$

The variable weight relations are

$$N_{1+\frac{1}{2}} = uE + (1-u)N_{1+\frac{1}{2}} \quad * u \geq 0 \quad (2a)$$

$$N_{j+\frac{1}{2}} = vE + (1-v)N_{j+\frac{1}{2}} \quad * v \geq 0 \quad (2b)$$

$$N_{m+\frac{1}{2}} = E - N_{m-\frac{1}{2}} \quad (2c)$$

where $1 \leq v, u \leq 2$ and are determined to insure positive coefficients of the spatial fluxes in the extrapolation for the edge fluxes. We now define the terms necessary to solve Eq. (1) by using Eq. (2), adopting the notation of the subroutines IN and OUT of TWO-TAN VW. Let

$$\begin{aligned} AA &= uA_{1+\frac{1}{2}}/YH(J) & u &\leq 0 \\ AC &= uA_{1-\frac{1}{2}}/YH(J) & u &\leq 0 \\ BS &= vS/YH(J) & v &\leq 0 \end{aligned} \quad (3)$$

$$CC = (A_{1+\frac{1}{2}} - A_{1-\frac{1}{2}})(G_{2+\frac{1}{2}} + G_{2-\frac{1}{2}})/(v_m YH(J))$$

$$CT = c_e v/YH(J).$$

Then the weights are calculated from

$$\begin{aligned} u &= 1/ANAX1(U1, 1.-AC/(CT + 2.*BS + CC))) \\ v &= 1/ANAX1(V1, 1.-BS/(CT + 2.*AA + CC)), \end{aligned} \quad (4)$$

where U1 and V1 are presently set to 0.5 in data statements in subroutines IN and OUT. If desired, these numbers can be changed to insure the use of weights u and v closer to unity (stop function scheme). After u and v are computed, AA and BS

are computed from

$$AA = AC - (1 - \alpha)AA \quad (5)$$

$$BB = \sqrt{2}AA.$$

Then, the cell centered flux is computed from

$$F = \frac{(AA^*F_{i+\frac{1}{2}} + BB^*F_{j+\frac{1}{2}} + CC^*F_{m-\frac{1}{2}} + QT)}{(AA + BB + CC + QT)}$$

$$i+\frac{1}{2} = 50$$

$$j+\frac{1}{2} = 750, \quad (6)$$

and the edge fluxes are extrapolated from Eq. (2).

In Eq. (6) QT is $V/\Delta t$ times the total source.

To initialize the angular flux $F_{m-\frac{1}{2}}$ the step function scheme, $F = F_{m-\frac{1}{2}}$, is used instead of Eq. (2c) which leads to an equation exactly like Eq. (6) except that CC = 0.

In (x,y) geometry, the above algorithm gives the same results as obtained with the TWOTRAN (x,y) program.⁵ In (r,z) and (r,θ) geometries, the scheme does not prevent negative fluxes because the angular flux is computed with the diamond scheme. However, very few negative scalar fluxes have been observed in extensive testing. In a severe shielding-type (r,z) benchmark problem,⁶ use of the above scheme gave consistently smoother fluxes than obtained with the diamond scheme, and, for relatively fine meshes, more accurate fluxes because oscillations in the diamond scheme results were suppressed. An example of TWOTRAN V_N flux errors in this problem is shown in Fig. 1.

The remaining oscillations in the scalar flux are due to ray effects.^{7,8} The modification described in the next section is designed to eliminate this kind of distortion.

III. CONVERSION OF DISCRETE ORDINATES EQUATIONS TO EQUATIONS LIKE SPHERICAL HARMONICS EQUATIONS - TWOTRAN IV

Because the discrete ordinates equation restricts particle streaming to the straight lines defined by the discrete directions, fluxes in low-order discrete ordinates approximations can be severely distorted, especially in problems where sources are isolated in absorbing media. This kind

Fig. 1. Percent deviation of the scalar flux from "exact" solution [approximate/exact = 1] $\times 100$ as a function of radius over the top edge of the cylinder ($z = 150$ cm). The LADL solution is an S_{16} discrete ordinates computation. The DET-10 result is also a discrete ordinates computation while the KAPL result is the combination of diffusion theory and Monte Carlo.

of distortion can be eliminated⁹ by adding a special source to the discrete ordinates equations. This source is chosen to transform the spherical harmonics moment equations of the discrete ordinates equations in the same way the analytic spherical harmonics equations would be transformed. The source can be chosen in other ways. In particular, Reed⁹ has shown how, in (x,y) geometry, the source can be selected to insure equivalence with spherical harmonics equations. We also describe an alternate source that has some possible programming advantages. We have created a version of TWOTRAN for a conventional truncation-type source and this alternate source. We discuss the theory of these two versions separately, but provide the programming changes for the first version only.

A. Theory of the Conventional Truncation Source -

TWO-TRAN PM

In a conventional P_L approximation, all the $L + 1$ order terms are removed from the L -order equations. This provides a consistent number of unknowns and equations. Why the $L + 1$ order moments can be removed from the L -order equation can be justified in a number of ways. It is frequently assumed that these moments (or their space derivatives) vanish. In the next section we try to create a set of discrete ordinate equations in which this is true. In this section we create a source which selectively removes the undesired moments (even though they may not vanish), following an observation made about the analytic spherical harmonics equations¹⁰ that indicates they are equivalent to the transport equation modified by a particular source. There are two choices for this source. In the plane geometry P_L equations, the source can be either (Ref. 10)

$$(L + 1)P_L \frac{d\phi_1}{dx}, \quad (7)$$

or (Ref. 11)

$$(L + 1)P_{L+1} \frac{d\phi_1}{dx}, \quad (8)$$

where ϕ_1 is the 1^{st} Legendre moment of the angular flux. The source of Eq. (7) enters only in the L -moment equation and eliminates the $L + 1$ moment term, truncating the set of equations. The source of Eq. (8) enters only the $L + 1$ moment equation and removes the L -order moment term, uncoupling the high-order equations from the low-order equations and leading to $\phi_L = 0$, $L \geq L + 1$, provided there are no true sources feeding these higher-moment equations. We began using a source of the form of Eq. (7) because our discrete quadrature could not correctly integrate spherical harmonic polynomials of too high an order. Because Eq. (7) is prefaced by a lower-order polynomial, it was easier to treat. Subsequent use of polynomials orthonormal with respect to an arbitrary discrete quadrature permitted sources like Eq. (8), of which the source of the next section is a special example, to be used.

Our objective in this section is to determine the $L + 1$ order terms that arise from forming the L -order moments of the discrete ordinate equations in

Eq. (1). We do this in general terms by first creating a set of orthonormal polynomials of order n , $U_n^k(\mu_n, \eta_n)$, $k = 0, 1, 2, \dots, N$, such that

$$\sum_{n=1}^{MT} w_n U_n^k(\mu_n, \eta_n) U_n^l(\mu_n, \eta_n) = \delta_{kl} \delta_{nn} \quad (9)$$

for whatever MT direction discrete ordinate quadrature set (μ_n, η_n, w_n) is being used. This is done in subroutine POLY described below. Our special source for Eq. (1) will be of the form

$$G_{1,0} = \sum_{k=0}^L U_L^k(\mu_n, \eta_n) w_{1,k}^k, \quad (10)$$

and will appear in only the $n = L$, $k = 0, 1, \dots, L$, moment equations formed by multiplying Eq. (1) by $w_n U_n^k(\mu_n, \eta_n)$ and summing. In particular, the balance equation will be unaffected for all $L > 0$, and $L = 0$ is not allowed by TWO-TRAN PM ($L = 1$ is used if $L = 0$ is accidentally selected).

The form of G is determined by finding all the $L + 1$ moment terms created when the left side of Eq. (1) is multiplied by $U_L^k(\mu_n, \eta_n)$. The first of these is due to the combination

$$w_n U_n^k(\mu_n, \eta_n) = \sum_{s=0}^{L+1} \sum_{t=0}^{L+1} r_{st}^k U_s^t(\mu_n, \eta_n). \quad (11)$$

Using the inner product notation

$$(h, g) = \sum_{n=1}^{MT} w_n h(\mu_n, \eta_n) g(\mu_n, \eta_n), \quad (12)$$

we determine the coefficients r_{st}^k from

$$r_{st}^k = (U_n^k, U_s^t) \quad \begin{aligned} k &= 0, 1, \dots, L \\ s &= 0, 1, \dots, L + 1 \\ t &= 0, 1, \dots, s. \end{aligned} \quad (13)$$

The $L + 1$ order terms of Eq. (11) are simply those terms on the right of Eq. (11) for which $s = L + 1$. It is convenient to collect these in $L + 1$ polynomials of $L + 1$ order:

$$U_{L+1}^k(\mu_n, \eta_n) = \sum_{t=0}^{L+1} r_{L+1,t}^k U_s^t(\mu_n, \eta_n) \quad \begin{aligned} k &= 0, 1, \dots, L. \end{aligned} \quad (14)$$

These polynomials are generated in subroutine POLY.
The presence of $\mathcal{L}_{L+1,1}^k$ due to the radial derivative

$$\mathcal{L}_{L+1,1}^k = (\mathcal{L}_{L+1,1}^k - \mathcal{L}_{L-1,1}^k) - \mathcal{L}_{L-1,1}^k. \quad (14)$$

is an exactly similar fashion we form

$$\mathcal{L}_{L+1,1}^k = \sum_{l=0}^{L+1} \mathcal{L}_{L+1,l}^k \mathcal{L}_{L+1,l}^k \quad (15)$$

where

$$\mathcal{L}_{L+1,l}^k = (\mathcal{L}_{L+1,l}^k - \mathcal{L}_{L-1,l}^k) \quad k = 0, 1, \dots, L$$

$$l = 0, 1, \dots, L+1, \quad (17)$$

and then form the axial portion of $\mathcal{L}_{L+1,1}^k$ from

$$\mathcal{L}_{L+1,1}^k|_{\text{axial}} = (\mathcal{L}_{L+1,1}^k - \mathcal{L}_{L-1,1}^k). \quad (18)$$

The remaining derivative term in Eq. (1), the angular curvature term, is now difficult to treat. We note that it involves a finite difference representation of an angular derivative. We must perform the analog of an integration by parts before the angular coefficient can be expanded. Ignoring the geometric term $(\mathcal{L}_{L+1,1}^k - \mathcal{L}_{L-1,1}^k)$, the $L+1$ moment of Eq. (1) gives

$$\sum_{n=1}^{M+1} \mathcal{L}_{L+1,n}^k (\mathcal{L}_{L+1,n}^k - \mathcal{L}_{L-1,n}^k) / \mathcal{L}_{L+1,n}^k. \quad (19)$$

We now use the identity

$$\mathcal{L}(\mathcal{L}) = \mathcal{L}\mathcal{L} + \mathcal{L}\mathcal{L}, \quad (20)$$

where

$$\mathcal{L}_n = (\mathcal{L}_{n+1/2} + \mathcal{L}_{n-1/2})/2$$

$$\mathcal{L}_n = (\mathcal{L}_{n+1/2} - \mathcal{L}_{n-1/2})/2 \quad (21)$$

$$\Delta\mathcal{L} = \mathcal{L}_{n+1/2} - \mathcal{L}_{n-1/2}$$

to rewrite Eq. (19) as

$$-\sum_{n=1}^{M+1} \mathcal{L}_{L+1,n}^k \mathcal{L}_{L+1,n}^k + \sum_{n=1}^{M+1} \mathcal{L}_{L+1,n}^k \Delta\mathcal{L} / \mathcal{L}_{L+1,n}^k. \quad (22)$$

The first term is obtained by noting that $\mathcal{L} = -\mathcal{L}_n$ by definition. We now define boundary values of $\mathcal{L}_n^k$ from

$$\mathcal{L}_n^k = (\mathcal{L}_{L+1,n+1/2}^k + \mathcal{L}_{L+1,n-1/2}^k)/2, \quad (23)$$

and by using Eq. (20) in the form

$$(\mathcal{L})\Delta\mathcal{L} = \Delta(\mathcal{L}\mathcal{L}) - \mathcal{L}\Delta(\mathcal{L}), \quad (24)$$

we rewrite Eq. (22) to give

$$-\sum_{n=1}^{M+1} \mathcal{L}_{L+1,n}^k \mathcal{L}_{L+1,n}^k - \sum_{n=1}^{M+1} \mathcal{L}_{L+1,n}^k \Delta(\mathcal{L}) / \mathcal{L}_{L+1,n}^k + \sum_{n=1}^{M+1} \Delta(\mathcal{L}\mathcal{L}). \quad (25)$$

We now note that the last sum yields only boundary terms and assert that these vanish. If true, this implies that

$$\sum_{n=1}^{M+1} \mathcal{L}_{L+1,n}^k \Delta(\mathcal{L}) / \mathcal{L}_{L+1,n}^k = - \sum_{n=1}^{M+1} \mathcal{L}_{L+1,n}^k [\mathcal{L}_{L+1,n}^k + \Delta(\mathcal{L}) / \mathcal{L}_{L+1,n}^k]. \quad (26)$$

We then determine the $L+1$ moment terms in the square bracket on the right of Eq. (26) just as we did previously for μ and η . This gives, including geometric coefficient,

$$\mathcal{L}_{L+1,1}^k|_{\text{curvature}} = -(\mathcal{L}_{L+1,1}^k [(\mathcal{L}_{L+1,1}^k - \mathcal{L}_{L-1,1}^k) \mathcal{L}_{L+1,1}^k]). \quad (27)$$

where the polynomials $\mathcal{L}$ are given by

$$\mathcal{L}_{L+1,1}^k(\mu_n, \eta_n) = \sum_{l=0}^{L+1} \mathcal{L}_{L+1,l}^k \mathcal{L}_{L+1,l}^k(\mu_n, \eta_n), \quad (28)$$

and

$$\mathcal{L}_{L+1,1}^k = (\mathcal{L}_{L+1,1}^k [1 + \Delta(\mathcal{L}) / \mathcal{L}_{L+1,1}^k]). \quad (29)$$

The values of $\Delta(\mathcal{L})$ are calculated by using Eq. (21) to generate boundary values of $\mathcal{L}$ recursively, that is, by using the definition of $\mathcal{L}$ in Eq. (21), from the recursion

$$\alpha_{m+\frac{1}{2}, L, m+\frac{1}{2}}^k = (\alpha_{m+\frac{1}{2}} + \alpha_{m-\frac{1}{2}}) U_{L, m}^k - \alpha_{m-\frac{1}{2}, L, m-\frac{1}{2}}^k. \quad (30)$$

We now examine several assumptions that have been made, tacitly or overtly. These are

1. The boundary terms in Eq. (25) vanish,
2. The diamond difference assumption for N_m of Eq. (21) applies, and
3. Numerical derivatives of the form $\Delta(U)/w_m$ for orders of U_n^k less than L do not give rise to moments of order $L+1$, i.e., no $L+1$ order moments occur in low-order moments of the discrete ordinates equations.

The first assumption was made in Eq. (25) to set the term

$$\sum_{n=1}^{M/2} \Delta(U_n) \quad (31)$$

to zero. The sum is taken on n levels for which the first and last α terms vanish. On such levels the first ΔU term can be assumed to vanish, starting the recursion of Eq. (30). However, we note that Eq. (30) is essentially a recursion for ΔU and that the last term on an n level need not vanish. Hence, the last term of Eq. (31) on such a level also need not vanish. This is tantamount to requiring extra terms of the form

$$(U_n)_{m+\frac{1}{2}} \Big|_m = \text{Last on each } n \text{ level} \quad (32)$$

in Eq. (26). For the diamond difference scheme at angle [Eq. (2c)], initialized with the step scheme, the last boundary angular flux on each n level can be written as a simple sum of the intermediate cell-centered fluxes so that Eq. (32) can be written as a function of n times N_m . This function can be tested for $L+1$ components by expansion just as u_m or v_m was tested. Such components were never found, but although plausibility arguments can be made for this result, no proof is available. The algorithm in TWOTRAN PH assumes that Eq. (32) does not contribute to the polynomials D .

To insure that the assumption $N_m = (N_{m+\frac{1}{2}} + N_{m-\frac{1}{2}})/2$ is valid, the diamond scheme is always used for the angle difference of the flux in TWOTRAN PH. No negative fixup is allowed. A variable weight spatial difference scheme is used with

$u = v$ in Eqs. (2a) and (2b) entered as input. The initialization of the angular flux solution with the step scheme, $N_{m+\frac{1}{2}} = N_m$ does not affect the right side of Eq. (2b) because the coefficient of N_m vanishes in these directions. For these directions $\alpha_{m-\frac{1}{2}} = 0$ and $\alpha_{m+\frac{1}{2}} = -w_m u_m$ so that the coefficient is

$$\begin{aligned} \alpha_{m+\frac{1}{2}, L, m}^k - \Delta(U_{L, m}^k) &= \\ \alpha_{m+\frac{1}{2}, L, m}^k - \alpha_{m+\frac{1}{2}, L, m+\frac{1}{2}}^k &= 0 \end{aligned} \quad (33)$$

because $U_{L, m}^k = U_{L, m+\frac{1}{2}}^k$ when $\alpha_{m-\frac{1}{2}} = 0$ from Eq. (30).

Finally, the numerical derivative of the function (U_n^k) does indeed give rise to high-order moments regardless of the order n . While in an analytic representation the derivative would reduce the order of the polynomial U_n^k by one, the numerical difference produces a function that requires all the orders of polynomials possible (in the finite discrete space) to represent. Apparently this defect of the present representation does not greatly affect the smoothness of solutions. The defect does not occur in (x, y) geometry where there is no curvature term ($A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}} = 0$).

The total special source added to the discrete ordinates equations is computed iteratively because it depends on the fluxes being calculated. It is added to the right hand side of Eq. (1) so that it will cancel the similar portion on the left and thereby eliminate the ray effect. The $L+1$ components of $\phi_{1, j}^k$ are generated during an iteration by forming the radial, axial, and curvature moments of Eqs. (15), (18), and (27). These are used during the next iteration to form the sum of Eq. (10), which is added to the total normal source. As will be seen below, the actual coding to implement this process is relatively simple.

Because the special source does not contribute to the balance equation, the iterative process for generating it is not accelerated by the coarse mesh rebalancing algorithms of TWOTRAN. Moreover, the process is relatively slow in converging. An attempt, with moderate success, was made to overrelax the process, but such an algorithm is not included in the changed code. Fortunately, most of the change due to the special source is effected during the first few iterations, even in problems with severe

distortions. It has been our practice to limit the number of inner iterations to 20 when using TWOTRAN PN.

B. Theory of an Alternate Truncation Source

To the right side of Eq. (1), we add a source of the form

$$C_{ij}^k = \sum_{n=0}^{L+1} U_{L+1}^k(\mu_n, \eta_n) C_{ij}^k, \quad (34)$$

where the $L+2$ source function components are given by

$$C_{ij}^k = (U_{L+1}^k, \mu \Delta(\Delta N) + \Delta \Delta(\Delta N)/\mu + \gamma \Delta N - \Delta N) \quad (35)$$

$$k = 0, 1, \dots, L+1.$$

This source is designed to make the U_{L+1}^k moment of Eq. (1) give the equation

$$\sigma_t V(U_{L+1}^k, N) = 0, \quad (36)$$

that is, to make the U_{L+1}^k moment of N vanish. Early experience with this kind of source had indicated a slow rate of convergence, but Carlson¹² noted that such a source could be generated recursively for iteration K from

$$(C_{ij}^k)_K = (C_{ij}^k)_{K-1} - \sigma_t V(U_{L+1}^k, N)_{K-1}. \quad (37)$$

This means that the same simple algorithm can be used for all geometries and that only the polynomials U_{L+1}^k need be created. These computational simplifications tend to offset the slower convergence.

C. General Remarks on Truncation Sources

Neither of the sources described above can create a system of equations equivalent to spherical harmonic equations for arbitrary order unless special conditions are met. Given a standard 2D S_N quadrature, there are more discrete ordinate equations than in the 2D P_{N-1} spherical harmonics system. In (x,y) geometry, depending on the precision of the discrete quadrature, equations equivalent to spherical harmonic equations can be formed by using the TWOTRAN PN source [Eq. (10)], but for the best quadrature now available,¹³ we believe the conversion $S_N - P_{N-3}$ is the highest possible in this fashion. For these inaccurate quadratures and the

conversion $S_N - P_{N-1}$, the functions U_N^k give rise to systems of moment equations that cannot be rearranged to have the same coefficients as the spherical harmonic equations. If a quadrature of sufficient precision were available, the conversion $S_N - P_{N-1}$ would be possible for this technique in (x,y) geometry, but would not be possible even then in curved geometry because of the numerical derivative difficulty.

In (x,y) geometry Reed⁹ and Jung et al.¹⁴ independently have shown how $S_N - P_{N-1}$ equivalence can be obtained regardless of the quadrature precision. They use the spherical harmonic polynomials and put an interpolating polynomial through the points defined by the discrete ordinate directions. Then all the integrals of inner products can be done analytically. Reed's source is of the form of Eq. (8) and, like the source of Eq. (34), is prefaced by a polynomial of order $L+1$. Interestingly enough, Reed observed that the rate of iterative convergence of his $S_N - P_{N-1}$ conversion depended on the degree of precision of the quadrature. In fact for too low a precision quadrature, divergence was observed.

Neither Reed or Jung developed a source for curved geometry. The numerical derivative remains a problem, although it is possible that the interpolatory polynomial could be used so that an analytic curvature derivative could be taken instead of using the alpha coefficients in the discrete ordinate equations.

D. Typical Results Obtained with TWOTRAN PN

In (x,y) geometry, results have been reported elsewhere.⁸ In a (r,z) shielding-type benchmark problem,⁶ TWOTRAN PN and the alternate source were used in an $S_8 - P_5$ conversion to compute the solution. The test problem consists of a spatially flat cylindrical source ($0 < r < 65$ cm, $-60 < z < 60$ cm) inside a homogeneous absorbing ($\sigma_a/\sigma_t = 0.16$ in the first group) cylinder approximately 13 mean free paths in radius and 28 mean free paths in height ($0 < r < 133$ cm, $-140 < z < 140$ cm). The percent deviation from the "exact" solution of the $S_8 - P_5$ conversion is shown in Fig. 2. Also shown is the same deviation for an S_{12} solution. Clearly the large "ray effect" distortions have been suppressed. The only other practical remedy is the use of a higher S_N quadrature. Here the maximum flux error

Fig. 2. Radial variation of the error in the scalar flux near the top of the cylinder. The S_{12} curve is the same as the LASL result shown in Fig. 1. Actual flux drops four orders of magnitude from cylinder center to top and another two orders of magnitude radially.

was reduced from 17 to 3% with an S_8 quadrature which has less than half as many directions as an S_{12} quadrature. The plot shown is near the top of the cylinder, where the largest errors in the whole system are observed. Smoother error plots with smaller errors were obtained for axial traverses near the outside of the cylinder. Here the effect of the curvature term is small (it is proportional to $1/r$) and results are similar to those obtained in (x,y) geometry. Errors for such an axial traverse are shown in Fig. 3.

Very similar results were obtained when the source of Eq. (34) was used. The radial error traverse for the same problem as above (Fig. 2) gave a maximum scalar flux error of 4.6% for the $S_8 - P_5$ conversion. In this case, the maximum error was at the center.

Our own preference has been for the use of TWO-TRAN PH rather than the alternate. We accept the fact that we are not solving equations exactly

Fig. 3. Axial variation in the scalar flux error for large r for the same problem as in Fig. 2.

equivalent to spherical harmonics equations. We do not expect to completely remove all ray-effect distortions. We do expect that almost all such effects are removed, at least so that the remaining distortion is of the order of the error made in the calculation itself. In return we believe that iteration is easier, and the algorithm works for any quadrature for $1 \leq L \leq 15N - 1$. Gelbard²⁵ maintains that solutions to the spherical harmonics equations are very slowly converging (with increasing order of approximation) near interfaces where the cross section is discontinuous. This is not true of S_n solutions. By using a hybrid, neither S_n nor P_L , we may be able to eliminate most of the flux distortion, but not acquire this P_L defect.

An unexplored option is the selective generation of the corrective source. In some systems, only the fast groups are subject to ray effects; there is no need for a corrective source in thermal groups. Similarly, it may not be necessary to compute or use the corrective source in source regions. A selective source insertion by space cell

is easily possible within the framework of the algorithm used in TWOTRAN PN.

E. Coding Changes to Convert TWOTRAN to TWOTRAN PN

The user of TWOTRAN is asked to specify two additional numbers as input: the integer $L + 1$ where L is the order of the spherical harmonic approximation desired, and a decimal value of u , a weight for the spatial difference scheme used in both Eq. (2a) and Eq. (2b). In the code, $L + 1$ is called IPN, and the range of L is such that $2 \leq L + 1 =$

IPN $\leq$ ISN. Also, in the code, u is called VMUT and the range of u is $1.0 \leq u = VMUT \leq 2.0$. These inputs are entered as shown in Fig. 4, a data input sheet which also explains some of the new options available with TWOTRAN. The value of VMUT is entered in the space indicated for SHUT.

New variables and first-word storage indicators in TWOTRAN PN are transmitted through a labeled common: /PN/. The contents of the common block are defined in Table I.

TABLE I
CONTENTS OF LABELED COMMON /PN/

Name	Pointer For Array	Remarks
LQ1	Q1(IPN,MM)	Polynomials U_n^k [Eq. (9)] evaluated for first sweep directions, $\mu < 0, \eta < 0$
LQ2	Q2(IPN,MM)	Same for $\mu > 0, \eta < 0$
LQ3	Q3(IPN,MM)	Same for $\mu < 0, \eta > 0$
LQ4	Q4(IPN,MM)	Same for $\mu > 0, \eta > 0$
LF1	F1(IPN,MM)	Polynomials F_{L+1}^k [Eq. (14)] evaluated for first sweep directions, $\mu < 0, \eta < 0$
LF2	F2(IPN,MM)	Same for $\mu > 0, \eta < 0$
LF3	F3(IPN,MM)	Same for $\mu < 0, \eta > 0$
LF4	F4(IPN,MM)	Same for $\mu > 0, \eta > 0$
LG1	G1(IPN,MM)	Polynomials G_{L+1}^k [Eq. (16)] evaluated for first sweep directions, $\mu < 0, \eta < 0$
LG2	G2(IPN,MM)	Same for $\mu > 0, \eta < 0$
LG3	G3(IPN,MM)	Same for $\mu < 0, \eta > 0$
LG4	G4(IPN,MM)	Same for $\mu > 0, \eta > 0$
LW1	W1(8*MM)	Double precision quadrature weights for all directions
LU1	U1(8*MM)	Double precision direction cosines μ for all directions
LV1	V1(8*MM)	Double precision direction cosines η for all directions
LA1J	A(IPN+1,IPN+1)	Coefficients of expansions made in subroutine POLY
IPN		Input order of spherical harmonic approximation plus one. $2 \leq IPN \leq ISN$
NPF		$NPF = ((IPN+1)(IPN+2))/2$ is the number of polynomials U
LR	R(IPN,IT)	Core storage of space dependent portion of special source -- prior iteration

TABLE I (Cont.)

LP	P(IPW,IT)	Core storage of space dependent portion of special source -- present iteration
LD1	D1(IPW,MM)	Polynomials D_{l+1}^k [Eq. (26)] evaluated at first sweep directions, $\mu < 0, \eta < 0$
LD2	D2(IPW,MM)	Same for $\mu > 0, \eta < 0$
LD3	D3(IPW,MM)	Same for $\mu < 0, \eta > 0$
LD4	D4(IPW,MM)	Same for $\mu > 0, \eta > 0$
LC1	C1(8*MM)	Double precision alpha coefficient used to evaluate recursion Eq. (30) in POLY
LC2	C2(8*MM)	Same
IPFP		ECS pointer of F array (IPW*IT*JT*MM words)
IPFR		ECS pointer of R array (IPW*IT*JT words)
IPMP		IPW+1
UT		Input variable weight $v = v$ in Eqs. (2a) and (2b)
UC		1. - UT

The common block /PW/ is used in the following subroutines:

TWOTRAN	UNDEF
INPUT11	IN
INPUT12	OUT
INPUT14	GRIND23
POLY	TESTS

These are the subroutines changed in TWOTRAN PW, and we now discuss the nature of these changes for each.

TWOTRAN Inclusion of labeled common only.

INPUT11 The line labeling output of the code is changed to indicate that the TWOTRAN PW version is used. Input statements are modified to read and print IPW and UT. MPF and the storage pointers of Table I are calculated. The polynomial storage pointers and related pointers are computed with the S_n and scatter coefficients, while LP and LD are added at the end of normal storage.

INPUT12 ECS storage for P is initialized to zero in this subroutine. The values IT and JT are required in this operation.

INPUT14 The call to subroutine POLY is made from INPUT14, after MPF*MM*8 temporary words of storage are set to zero beginning at the LP1

pointer. These words are cleared again after return from POLY.

POLY This routine is the major addition to TWOTRAN in the PW version. It generates the U, F, G, and D polynomials. For high-order quadratures ($S_n, N > 12$) some lack of precision was noted in the orthonormality of the polynomials. This difficulty was eliminated by using a few temporary double precision variables. Because TWOTRAN uses only one octant of the directions ($\mu, \eta > 0$), it was convenient in POLY to generate direction sets covering all four octants needed. Thus, after normalizing the quadrature weights to unity, a complete set of weights, direction cosines, and curvature coefficients is generated for all 4π directions. With this set, the orthonormal polynomials U_n^k are generated from the following algorithm. Let $\tilde{U}$ denote the unnormalized polynomial. Then U_n^k is constructed as a sum of normalized polynomials of lower degree plus a combination of powers of μ and η of degree n

$$\tilde{U}_n^k = \sum_{i=0}^{n-1} \sum_{j=0}^i b_{ij} U_i^j + \mu^{n-k} \eta^k. \quad (30)$$

[illegible][illegible]

11

The coefficients b_{ij} are generated from

$$b_{ij} = -(U_i^j, \mu^{n-k} \eta^k) \quad j = 0, 1, \dots, l$$

$$i = 0, 1, \dots, n-1$$

$$= - \sum_{m=1}^{N_k} w_m U_i^j(\mu_m, \eta_m) \mu_m^{n-k} \eta_m^k. \quad (39)$$

In POLY, the b coefficients are stored as a vector and the double sum in Eq. (38) is replaced by a single sum. Once the polynomial is formed, it is normalized by the factor $1/T$ where

$$T = (\bar{U}_n^k, \bar{U}_n^k)^{\frac{1}{2}}, \quad (40)$$

provided $T > 10^{-6}$. This precaution is necessary because there are more $L+1$ order polynomials than there are directions in standard S_n sets. In fact there are $(L+2)(L+3)/2$ polynomials of order $L+1$ and $N(N+2)/2$ directions in an S_n set. For $N=2$ there are four directions and six polynomials for $L=1$. When one attempts the above orthonormalization for this set of parameters, one finds that there are two polynomials that are identically zero. In the general case for $IPN = L+1 = INW$, we always found $INW/2$ zero polynomials.

The orthonormality of the polynomials formed is checked and an informational diagnostic is printed for polynomials with inner product sums larger than 10^{-12} , including those for which the sum should be unity.

Once the polynomials U are available, the polynomials F and G are generated exactly as indicated in Eqs. (13), (14), (16), and (17) with the array $A(IPN+1, IPN+1)$ being used to store the f and g coefficients. When F is generated it is stored in the (single precision) arrays $F1$, $F2$, $F3$, and $F4$. Similarly, G is computed and stored.

The treatment of S is complicated somewhat by the necessity of generating recursively the boundary values of U [Eq. (30)], but proceeds in essentially the same fashion.

An extra factor w_m is included in F , G , and D to save a multiplication in subroutines IN and OUT .

INNER At the beginning of **INNER**, the ECS storage index of F is computed. Then, at the beginning of an inner iteration, values of F are read from ECS into the K array and the N array is

cleared. This is done for each j level, that is, for each vertical cell, of the system. During the iteration the new special source is accumulated in P , while the old special source is used from R . After the completion of the **IN-OUT** traverse for downward directions on a j level, the partially accumulated new source is stored in ECS as is the old source. When calculation begins on upward directions, for each j level, P and R are refreshed from ECS. When the **IN-OUT** traverse has been finished on these levels, the inner iteration is finished and only P is stored in ECS. In this portion of **INNER**, the calls to routines **IN** and **OUT** are modified to transmit R , P , U , F , G , D and necessary indices and parameters.

After inner iteration is performed, rebalance factors are computed and these are applied to P as well as other quantities.

IN This routine is modified to dimension all the new quantities called RR , FF , QQ (7), FF , GG , and DD to prevent duplication of FORTRAN variables. At the beginning of the calculation, the source is augmented by the addition

$$QT = QT + \sum_{k=1}^{IPN} QQ_{k,n} wR_{k,i}, \quad (41)$$

where the sum here is that of Eq. (10). The calculation of new fluxes is altered to use the variable weight difference scheme and no negative flux fixup is allowed. When the new edge fluxes are computed and before they are stored, the inner products of Eqs. (15), (18), and (27) are accumulated in the form

$$FF_{k,i} = \sum_n FF_{k,n} * IX + GG_{k,n} * TY + DD_{k,n} * TAP$$

$$k = 1, 2, \dots, IPN, \quad (42)$$

where

$$IX = A_1(I+1)w_{i+\frac{1}{2}} - A_1(I)w_{i-\frac{1}{2}} \quad (43a)$$

$$TY = N_1 A_3(I) (N_{j+\frac{1}{2}} - N_{j-\frac{1}{2}}) \quad (43b)$$

$$TAP = -A_2(I)w_i. \quad (43c)$$

In these equations, the geometric factors are divided by the common $YH(J)$ factor. In Eq. (43b) the sign of EA is adjusted to make sure that $EA^*(BV(1,M) - TJ) = |EA|(N_{j+\frac{1}{2}} - N_{j-\frac{1}{2}})$ whether upward or downward directions are being treated.

OUT Changes in OUT are almost identical to those in IN.

GRIND23 The calling sequence to TESTS is altered.

TESTS The P array is normalized by the over-all-groups-rebalance factors. Whether or not this is essential has not been investigated. Without the availability of a large rapid access storage unit, it would be faster not to normalize P. To test this possibility the changes in GRIND23 and TESTS could simply be omitted.

A few minor changes are necessary to dump the P array and the PN common from storage for use in restart dumps. The changes and auxiliary routines used in this process are the same as those described in the next section.

IV. ADDITION OF A FIRST COLLISION SOURCE IN CYLINDRICAL GEOMETRY - TWOTRAN FC

If an inhomogeneous source problem is being solved, ray effects can be mitigated by calculating the uncollided flux analytically and then using this flux to compute the much less localized source of first collisions for a normal E_g calculation. In this section we describe the modifications to TWOTRAN necessary for the calculation of the uncollided flux from a point source on the axis of a cylinder. Designed for air burst calculations, the modification is made to permit a pointwise variation of the system material density in the axial direction. However, because of the difficulty in calculating the optical path, the system must be homogeneous in the radial direction. In addition only vacuum exterior boundary conditions are allowed for the cylinder.

The calculation of the uncollided flux is simple; the major modifications for the use of a first collision source are those necessary to maintain conservation, and those necessary to store and manipulate two sets of fluxes. Conservation is important because errors are made in adapting an analytic solution to a finite mesh. For example, we take the uncollided flux to be the flux at the

center of a discrete cell, not the average of the analytic solution over the cell. We therefore calculate the total number of collisions and leakage due to the analytic solution and normalize the uncollided flux so that this number is equal to the input source in each group.

A. General Theory

Let the total flux be divided into an uncollided and a collided portion

$$\phi = \phi^{uc} + \phi^c. \quad (44)$$

The equations satisfied by these two fluxes are, for a source height z_0 ,

$$v \cdot \nabla \phi_g^{uc} + \sigma_{tg} \phi_g^{uc} = \frac{Q_g \delta(z) \delta(z - z_0)}{2\pi}, \quad (45)$$

$$v \cdot \nabla \phi_g^c + \sigma_{tg} \phi_g^c = \sum_{h=1}^{NGM} \sum_{n=0}^{NCT} (2n+1) f(z) \sigma_{n,h} \sum_{l=0}^n R_{n,l}^h \left(\phi_{n,h}^{uc} + \phi_{n,h}^c \right). \quad (46)$$

The notation in these equations is the same as in the TWOTRAN manual¹ except that $f(z)$ represents the pointwise axial density variation of cross sections. A user of TWOTRAN FC enters a source spectrum Q_g and the uncollided flux ϕ_g^{uc} is computed, normalized, and stored by subroutine VIRGIN. This flux is used subsequently to compute the source in Eq. (46), which is solved in the usual fashion.

The solution to Eq. (45) is

$$\phi_g = \frac{Q_g e^{-\tau(r,z)}}{4\pi r^2} \ell(\mu_{sph} - 1), \quad (47)$$

where

$$r^2 = \rho^2 + z^2$$

$$v(z) = \int_{z_0}^z \sigma_t f(z) dz / (z - z_0), \quad (48)$$

and μ_{sph} is a direction cosine measured from the radius vector from the source point to the field point. Subroutine VIRGIN calculates $v(z)$ for z values corresponding to cell midpoints and the top

and bottom cylinder boundaries. It then proceeds through the system evaluating Eq. (47). At any point (ρ, z) , μ_{sph} corresponds to

$$\mu_{cyl} = \rho/r$$

$$\eta_{cyl} = (z - z_0)/r, \quad (49)$$

but these cosines may not be part of the set used in the $\hat{g}_n$ calculation. The angular flux is then assigned to the "closest" available direction. This approximation is not serious since only the spherical harmonic moments of ϕ_g^{uc} are used in Eq. (46) and this "closest" direction is used only to evaluate the spherical harmonic polynomial P_n^k . However, when $ICST > 0$ and is small, the spherical harmonic expansion of ϕ_g^{uc} may be negative in certain directions because the analytic flux is so highly collimated. This leads to a negative collided flux source in Eq. (46) and sometimes to negative total scalar fluxes.

3. Code Changes

We now describe the nature of the changes made to implement the first collision source in TWOTRAN. Surprisingly, there are more of these than for the PN version mostly because of the complexity of the particle balance tables with two complete sets of fluxes. The altered program is designed to run other than first collision source problems and obtain the same results as the unaltered version. However, the present version of TWOTRAN PC does not permit a distributed source in addition to the point source on the cylinder axis, although only minor changes would be required to allow this option.

The TWOTRAN PC user supplies three additional inputs besides regular TWOTRAN input. These are IQC (0 or 1), a no or yes trigger indicating the input of an uncollided flux spectrum; SHPT, the height on the axis of the point source; and IGM numbers indicating the amplitude of the source in each group. These numbers are submitted after all other input and any one, but not all, of them may be zero.

Two special labeled commons, /COLLIDE/ and /VIRGIC/, are used. the /VIRGIC/ common is used to pass information between VIRGIN and GROUP, two new subroutines, and also appears in TWOTRAN and AIDCMP.

The common /COLLIDE/ appears in TWOTRAN, AIDCMP, INPUT11, INPUT15, GRIND21, VIRGIN, GROUP, INITQ, OUTER, COBMS, YENTS, and OUTPUT3. The contents of these commons are defined in Tables II and III. We now discuss the changes made in each subroutine.

TWOTRAN Inclusion of labeled commons only.

AIDCMP This is a special subroutine used to dump the labeled commons and uncollided flux for restart and final dumps. Called by ECHUMP, the routine calls TASDMP to write the labeled common blocks.

INPUT11 The name of the program is changed to TWOTRAN PC for output labeling. IQC and SHPT are read and printed, the pointer IQC is calculated, and the ECS pointers ICPTRL and ICPTRD and lengths ICLENL and ICLENR are computed.

INPUT15 The array QC(IGM) is read.

GRIND21 After INITIAL is called, the routine VIRGIC is called to insure that the right, bottom, and top boundary conditions are vacuum conditions, that the geometry is (ρ, z) , and that there is no radial cross section variation. Then VIRGIN is called for each group to generate the uncollided flux and GROUP is called to initialize the system-wide balance table entries. Finally, if $ICST \neq 0$, all sources are normalized. In GRIND21 the Q array is used for core storage of the uncollided flux.

VIRGIN First, in this subroutine, the product $q_j(z)$ is generated as the function EL_j , $EL_{j+1/2}$, $j = 1, JT$, and EL_{JT+1} for the bottom, cell midpoints, and top of the system. Then, the μ boundaries of the direction set used are determined for use in calculation of the direction "closest" to that defined by μ_{cyl} and η_{cyl} . A symmetric direction set is assumed so that the η boundaries are the same as the μ boundaries. Next, for every cell in the system, the uncollided flux is computed and its spherical harmonic moments formed. At the same time, the right (RL), bottom (BL), and top (TL) leakages, and the total collision rate (TC), absorption rate (TA), and self-scatter (TSS) rate are accumulated. Then these quantities and the flux components are normalized by multiplication by the factor

$$T = QC/(TC + RL + BL + TL), \quad (50)$$

where QC is the magnitude of the group source. The normalized uncollided flux is written in ECS in GRIND21.

TABLE II
CONTENTS OF LABELED COMMON /FIRTC/

<u>Name</u>	<u>Remarks</u>
SNRT	Input source height
TC	For group g, the total number of particles colliding in the uncollided flux
TA	Total number of uncollided absorptions, group g
TSE	Total number of uncollided self-scatters, group g
TL	Total uncollided top leakage, group g
RL	Total uncollided right leakage, group g
BL	Total uncollided bottom leakage, group g

TABLE III
CONTENTS OF LABELED COMMON /COLLIDE/

<u>Name</u>	<u>Pointer for Array</u>	<u>Remarks</u>
LQC	QC(IOM)	Input source spectrum
IQC		Input uncollided flux trigger
ICPEXL		ECS pointer for uncollided flux array
ICPTED		ECS pointer for balance table arrays
ICLPEL		ECS length of uncollided flux (NBLJ*IES4)
ICLTED		ECS length of balance table arrays (11*TOP)
LEQC		Core pointer for a vector that contains a zero if the uncollided flux is zero in that group. Vector is IOM long.
LZPX		Core pointer for a vector that contains a zero if the flux is zero in that group. Vector is IOM long.
LZXS		Core pointer for a vector that contains a zero if all cross sections of materials used in the computation are zero in that cross section table row. Vector is IOM long.
IFCDEN ^a		Random disk indicator of the uncollided flux block (0/N = IOM contained/number of tracks required for a group).

^aUsed in random disk TMOEAM.

GROUP This routine initializes some of the eleven arrays in the system-wide balance tables. No effort is made to adjust the coarse mesh balance tables. In **GROUP** the source strength is added to $QG(G)$, TSR is added to $SS(G)$, TA to $ABG(G)$, RL to $RL(G)$, RL to $HL(G)$, TL to $TL(G)$ and $TL + RL$ to $VL(G)$. All of the groups in the balance table are written into MCS in **GRIND21**.

Subroutines **VINGIN** and **GROUP** are called for each group until the entire uncollided flux and corresponding balance table items have been computed and stored. This operation is performed only once at the beginning of a calculation.

INITG Normalization of any regular input source is skipped.

CUTER If $IQC \neq 0$, the balance tables generated in **GROUP** are read from MCS at the beginning of each outer to initialize the arrays QG , PG , SIN , SS , $SCUT$, HL , VL , TL , RL , and ABG . Then, the collided flux is calculated by using the same algorithm used to calculate the scattering source to the group except that self-scatter is not skipped. At this point, the uncollided flux is read into the $FLUX$ array and the total source to a group is calculated in Q . When $IQC \neq 0$, the inscatter into a group is computed from $SIN(G) = TS - PG(G) - SS(G)$ where $SS(G)$ is due to the uncollided flux only and plays the same role as $QG(G)$ does in a normal **TWO-TRAN** problem.

Several other changes are made in **CUTER** to prevent the zero initialization of SIN , SS , ABG , and $SCUT$ when there is no source in a group and SS , ABG , $SCUT$ when there is a source. Combinations of no sources of any kind in a group and sources is purely absorbing groups have been executed successfully with **TWO-TRAN** PC.

GROUP The zero initialization of HL , RL , VL , TL , RL , HAL , and the test $IF(TS.LE.0.)GO TO 123$ are skipped if $IQC \neq 0$. A test is made to prevent division by zero (which can happen in groups with no source) in the calculation of $HAL(G)$.

TEG2 Because no attempt was made to calculate coarse mesh rebalance flows for the uncollided flux, no coarse mesh rebalance factors can be applied. Instead, a single factor, calculated from the total source divided by the total leakage and total absorption, is used.

OUTPUT3 The uncollided and collided fluxes are added together for use in all output prints, plots, etc.

XRCK To minimize reading of fluxes and sources, it is convenient to check and skip reading of zero quantities. If all the cross sections in the system are zero for a row in the cross section table, then the flux need not be read to compute the scattering source to the group. **XRCK** checks for this possibility and calls **XRCK** to do the actual checking. Similar tests are made for the uncollided and collided flux arrays.

C. Typical Results

Figure 5 shows the scalar flux along the outside edge of a cylinder in two approximations for a source on the axis in a purely absorbing cylinder. One curve is the normal **TWO-TRAN** S_0 flux showing a severe ray effect, and the other curve is the uncollided flux from **TWO-TRAN** PC. Figure 6 shows the same comparison when a small amount of scattering is present. Now the **TWO-TRAN** PC result shows a small ray effect due to the collided flux S_0 computation. This is not unexpected because the first collision source from a point source is still localized about the point. Nevertheless, the solution is much improved compared to the standard **TWO-TRAN** result because most of the flux is the accurately computed uncollided flux. As the scattering ratio increases, less of the total flux is uncollided, but the ray effect in the collided flux is also less severe.

Fig. 5. Comparison of uncollided scalar edge flux from **TWO-TRAN** PC with normal **TWO-TRAN** flux for problem shown.

Fig. 6. Comparison of TWOTRAN PC scalar edge flux with normal TWOTRAN flux for problem shown.

V. TWOTRAN STRUCTURE CHANGES

In this section, we describe the current structure of TWOTRAN by reproducing several tables describing labeled common blocks, overlay structure, blank common, etc. Changes indicated for the PC or FC version, but not described previously are related to the use of random disk storage.

A. Overlay and Common Changes

Changes in overlays and common blocks are shown in Tables IV through IX.

TABLE IV

OVERLAY STRUCTURE OF THE TWOTRAN PROGRAM^a

<u>OVERLAY (0.0)</u>	<u>OVERLAY (1.0)</u>	<u>OVERLAY (2.0)</u>	<u>OVERLAY (3.0)</u>
INITIAL	INITIAL	INITIAL	INITIAL
SCALAR	SCALAR	SCALAR	SCALAR
BLANK			
CLERK	<u>OVERLAY (1.1)</u>	<u>OVERLAY (2.1)</u>	<u>OVERLAY (3.1)</u>
WRITE	WRITE	WRITE	WRITE
RECHECK			
RECHECK(PC)	<u>OVERLAY (1.2)</u>	<u>OVERLAY (2.2)</u>	<u>OVERLAY (3.2)</u>
RECHECK	RECHECK	RECHECK	RECHECK
ADJUST(PC,FC)	ADJUST	ADJUST	ADJUST
RECHECK(PC,FC)	LAKE	LAKE	LAKE
TANDEM(PC,FC)			
RECHECK(PC)	<u>OVERLAY (1.3)</u>	<u>OVERLAY (2.3)</u>	<u>OVERLAY (3.3)</u>
WRITE	WRITE	WRITE	WRITE
RECHECK	RECHECK	RECHECK	RECHECK
FORWARD			
BACKWARD	<u>OVERLAY (1.4)</u>	<u>OVERLAY (2.4)</u>	<u>OVERLAY (3.4)</u>
RECHECK	RECHECK	RECHECK	RECHECK
LAKE	LAKE	LAKE	LAKE
FORWARD			
	<u>OVERLAY (1.5)</u>	<u>OVERLAY (2.5)</u>	<u>OVERLAY (3.5)</u>
	RECHECK	RECHECK	RECHECK
	LAKE	LAKE	LAKE
	<u>OVERLAY (1.6)</u>	<u>OVERLAY (2.6)</u>	<u>OVERLAY (3.6)</u>
	RECHECK	RECHECK	RECHECK
	LAKE	LAKE	LAKE
	<u>OVERLAY (1.7)</u>	<u>OVERLAY (2.7)</u>	<u>OVERLAY (3.7)</u>
	RECHECK	RECHECK	RECHECK
	LAKE	LAKE	LAKE
	<u>OVERLAY (1.8)</u>	<u>OVERLAY (2.8)</u>	<u>OVERLAY (3.8)</u>
	RECHECK	RECHECK	RECHECK
	LAKE	LAKE	LAKE
	<u>OVERLAY (1.9)</u>	<u>OVERLAY (2.9)</u>	<u>OVERLAY (3.9)</u>
	RECHECK	RECHECK	RECHECK
	LAKE	LAKE	LAKE

^a(PC) is a routine used by TWOTRAN PC but not by TWOTRAN. (FC) is a routine used by TWOTRAN FC but not by TWOTRAN.

^bUsed in random disk storage.

The contents of common block IA have been slightly modified and these modifications are indicated in the table below.

TABLE V
COMMON BLOCK IA CHANGES

Position	Name	Pointer for Array	Remarks
35	LDGF		Density factor indicator
97	IXDP	XDP(IT)	Radial fine-mesh density factors (conditional input on ISDF.EQ.1)
98	LYDP	YDP(JT)	Axial fine-mesh density factors (conditional input on ISDF.EQ.1)
429	SLIMIT		Rebalance counter for INNER
748	MCARDS		PIDO cross section card input indicator
750	IOLYCS		Overlay indicator as a two-digit octal number

TABLE VI
CONTENTS OF NAMED COMMON BLOCK FWBGN1

The named common block FWBGN1 contains the information required to continue the processing of the current problem if it is restarted.

Name	Contents and Remarks
IDUSE	Last title card. This is a vector, 12 words in length, used for the title of the plot routines. The words are A6 in format.
LAST	Length of common block A used by the current problem.
LASTEX	Length of LCN used by the current problem.
TIMACC	Accumulated problem running time.
IGROUP	Group number of restart dump.
IPCO	LCN pointer of the source to the group block. This is the source to the group calculated by OUTER and used in INNER. This block is used for parallel reading of the flux block in the computation of the source to a group in OUTER when the flux is stored on random disk.
LTSC	Length of the source to the group block.
IPFL	LCN pointer for the first group of the flow block. The flow block contains the coarse-mesh partial currents matrices for the upward, downward, rightward, and leftward directions.
LTFL	Length of the flow block for a group.

TABLE VI (Cont.)

Name	Contents and Remarks
IPFX	LCM pointer for the first group of the flux block. The flux block contains the three-dimensional flux array as well as boundary arrays stored consecutively. When the flux block is stored on random disk, the LCM area is only one group in length and is used for reading and writing of this block.
LTFX	Length of the flux block for a group.
LTFX	Length of the three-dimensional flux array.
IPXDSK ^a	Random disk indicator of the flux block ($0/N$ = LCM contained/number of tracks required for a group). One track contains 512 words.
IPXS	LCM pointer for the first group of the cross section block. The cross section block contains the cross section array, the adjoint absorption vector (when problem is adjoint), and the spatial total cross section matrix. When the cross section block is stored on random disk, the LCM area is only one group in length and is used for reading and writing of this block.
IPXSCT	LCM pointer for the first group of the spatial total cross section matrix within the cross section block.
LTXS	Length of the cross section block for a group.
LTXS	Length of the cross section array for a group.
LTXS	Length of the cross section array and the adjoint vector for a group.
IXSDSK ^a	Random disk indicator of the cross section block ($0/N$ = LCM contained/number of tracks required for a group).
IPQS	LCM pointer for the first group of the Q-source block. The Q-source block contains the Q-source array and associated boundary source matrices (right, top, and bottom). The block is present only when a Q calculation is requested. The associated boundary matrices are present within the block only when each boundary source is requested. When the Q-source block is stored on random disk, the LCM area is only one group in length and is used for reading and writing of this block.
LTQS	Length of the Q-source block for a group.
IQSDSK ^a	Random disk indicator of the Q-source block ($\sim 1/0/N$ = none/LCM contained/number of tracks required for a group).
IERRC	LCM parity error indicator.
I2	Final flux print indicator ($0/1/2$ = all/isotropic/none).
I4	Final fission print indicator ($0/1$ = yes/no).
I6	Coarse-mesh balance table print indicator ($0/1$ = yes/no). Caution -- an extra OUTER iteration is required to print these tables.
NLINE	Number of contour lines to be used ($\sim N/0/N$ = log of flux for plot with N lines/flux for plot with 25 lines/flux for plot with N lines). The number of plot lines must be fifty or less.

TABLE VI (Cont.)

Name	Contents and Remarks
IPLOT	Type of flux film plot desired (0/1/2/3 = none/contour/projection/both film plots).
ISPARG	Last word address of the Q-source block in core.

^aUsed in random disk TWOTRAN.

TABLE VII

CONTENTS OF NAMED COMMON BLOCK FWGCH2

The named common block FWGCH2 consists primarily of those indicators used by the program but not vital to restart. Parameters which define options are set in program TWOTRAN.

Name	Contents and Remarks
MAXLEN	Maximum length of the main common data block A.
MAXSCS	Maximum length of the LCM storage available to the problem. This is obtained from the job card.
LESCCB	Length of the named common block FWGCH1 which must be saved for restarts.
LEPCIA	Length of the common parameter block IA which must be saved for restarts.
IFSOVY	Overlay file name. The file name is given in a left justified, zero fill Hollerith form and is not entered in the file name table of the program card.
IRCOVY	Recall overlay indicator. When the overlay is in core, it will be reloaded from the overlay file only if the indicator is set properly (0/6 HRECALL = no/yes).
TIMDOP	Minimum time between periodic dumps (C/F = no periodic dumps/seconds).
TIMSLD	Elapsed time since last dump.
TIMOFF	Floating-point form of the input fixed-point parameter IFLIM.
IFILM	File unit number of the SC-4020 plot film output.
ISUPLB	Cross section library sigma up indicator (0/1 = sigma up term <u>NOT</u> contained in cross sections/ sigma up present in upscatter cross sections).
I1	Full flux guess input print indicator (0/1 = yes/no). The indicator is effective only when ISTART is two or minus two.
I3	Cross section print indicator (0/1/2 = all/mixed/none).
I5	Normalised source print indicator (0/1 = yes/no).
IFITAP ^a	File name (left justified, zero fill) of the flux block when used on random disk.
IFPINT ^a	Initial logical track number of the flux block.
IFCOLT ^a	Maximum number of tracks allowed per group for the flux block.

TABLE VII (Cont.)

Name	Contents and Remarks
IXSTAP ^a	File name (left justified, zero fill) of the cross section block when used on random disk.
IXSINT ^a	Initial logical track number of the cross section block.
IXSDLT ^a	Maximum number of tracks allowed per group for the cross section block.
IQSTAP ^a	File name (left justified, zero fill) of the Q-source block when used on random disk.
IQSINT ^a	Initial logical track number of the Q-source block.
IQSDLT ^a	Maximum number of tracks allowed per group for the Q-source block.
IMODER	Mode error information. This is a mode error by-pass vector of two words. The first word contains the transfer address to be taken in the event of an arithmetic mode error. The second word at the time of a mode error will be filled with the error information.
XPSET1 ^a	Monitor communication block for the random disk of the flux block. It is a vector of twenty words.
XPSET2 ^a	Monitor communication block for the random disk of the cross section block. It is a vector of twenty words.
XPSET3 ^a	Monitor communication block for the random disk of the Q-source block. It is a vector of twenty words.

^aUsed in random disk TWOTRAN.

TABLE VIII
CONTENTS OF NAMED COMMON BLOCK LOCAL

The named common block LOCAL contains information that is passed from overlay to overlay for a problem but is not needed in restart.

Name	Contents and Remarks
NERROR	Parameter input-error indicator.
ITLIM	Fixed-point time problem removal value (O/N = no/seconds).
ISNT	S _n library request indicator.
MCR	Number of nuclides requested from cards.
MTP	Number of nuclides requested from library.
LCPT	Core pointer for the library ID request table. Block is 2*MTP in length.
IERRET	Storage error indicator.
IERRST	Restart error indicator.

TABLE VIII (Cont.)

Name	Contents and Remarks
NEXTER	Fetch next case indicator. If a factor occurs after all input is successfully read, next problem is fetched.
NEXTRA	File unit number of a scratch edit file.
JFIOC	Return indicator used in overlay.
NEEDIT	File unit number upon which edit input information is placed.
EDIT	Edit information. This is a vector of two words that contains the edit values NE and $NOFME$.
LIMIT	Storage length required by cross section phase of input.
LENCLR	Length of the partial block to be cleared during input.
LEPRMR	Core pointer of fine radial mesh for plot.
LEPRMY	Core pointer of fine axial mesh for plot.
LEPTMA	Core pointer of first temporary block for plot. The length of this block is the maximum of the total number of fine radial mesh intervals, the total number of fine axial mesh intervals, or 21.
LEPTMB	Core pointer of second temporary block for plot. The length of this block is the same as that of the first temporary plot block.
LEPTMC	Core pointer of third temporary block for plot. The length of this block is the same as that of the first temporary plot block.
LEPTMD	Core pointer of fourth temporary block for plot. The length of this block is the same as that of the first temporary plot block.
LEPTFL	Core pointer of scalar flux block for plot.
ICFLOG	Contour plot log flux indicator (0/1 = no/yes).
ICPIOT	Contour plot indicator (0/1 = no/yes).
IPFIOT	Projection plot indicator (0/1 = no/yes).

TABLE IX

CONTENTS OF NAMED COMMON BLOCK COLLDT^a

The named common block COLLDT is used only in TWOTRAN PC. The contents of the block are not needed for a restart.

Name	Contents and Remarks
IFCFLP	File name (left justified, zero fill) of the uncollided flux block when used on random disk.
IFCINT	Initial logical track number of the uncollided flux block.
IFCMLS	Maximum number of tracks per group for the uncollided flux block.
XPART ^b	Monitor communication block for the random disk of the uncollided flux block. It is a vector of twenty words.

^aUsed in random disk TWOTRAN.

B. Flux Dumps and Problem Restart

The restart procedure for continuation of a problem remains the same. The fixed point and floating point parameter entries which may be changed by using the namelist feature and the physical format of the dump tape have been revised.

TABLE X
NAMELIST RESTART VALUES

The integers that may be changed upon restart are

1.	IIYL	Maximum number of inner iterations.
2.	ITLIM	Time limit.
3.	IEDOPT	Edit options.
4.	I2	Final flux print indicator.
5.	I4	Final fission print indicator.
6.	I6	Coarse-mesh balance table print indicator.
7.	ELINE	Number of contour lines to be used in plot.
8.	IPLOT	Type of flux file plot desired.

The floating-point numbers that may be changed are

1.	EV	Eigenvalue guess.
2.	EVN	Eigenvalue modifier.
3.	FWK	Parametric value of k_{eff} or alpha.
4.	KLAL	Search lambda lower limit.
5.	KLAM	Search lambda upper limit.
6.	ELAX	Fine-mesh search precision.
7.	EPS	Convergence precision.
8.	POD	New parameter modifier.

The first record of the restart dump tape consists of vital information needed for reading the restart tape. The record consists of 17 words ordered as follows: (1) number of groups (IGN), (2) number of flux components (IN), (3) product of total number of fine radial mesh intervals and total number of fine axial mesh intervals (ITOT), (4) length of permanent named common block PWGHE1 (LENMCH), (5) length of parameter common block IA (LENCTA), (6) length of data common block A used by the problem (IAST), (7) LCM pointer for the first group of the flux block (IPFX), (8) length of the flux block for a group (LFPX), (9) LCM pointer for the first group of the cross section block (IPXS), (10) length of the cross section block for a group (LXPX), (11) LCM pointer for the first group of the flow block (IPFL), (12) length of the flow block for a group (LFXL), (13) LCM pointer for the first group of the Q-source block (IPQS), (14) length of the Q-source block for a group (LQSL), (15) flux block random disk indicator (IPRDX), (16) cross section random disk indicator (IPRDX), and (17) Q-source random disk indicator (IPRDX).

The second record consists of the permanent named common block PWGHE1, the third record consists of the permanent parameter block IA, and the fourth record consists of the main common data block A.

Following the main common data block record is a series of flux records, a series of cross-section records, a series of Q-source records (present only when a Q-source calculation is being made), and a series of flow records.

A series of records is composed of a pair of records for each group beginning with group one and proceeding until the last group. The first record of a pair is a single entry specifying the form of the second record (0/1 = all entries equal/entries not equal). The second record contains either a single entry or all the data.

C. Additions to the Restart Tape

1. TRUTHAN PC. After the last record of the standard TRUTHAN dump tape, another set of records is added. The first record consists of the named

common block COLLIDE, and the second record consists of the named common block FIRSTC. The next records consist of a series of records containing the group dependent uncollided flux block. The last two records contain the neutron balance table information. Both these series have the same format as is used for other group dependent records.

2. TWOTRAN PW. After the last record of the

standard TWOTRAN dump tape, another set of records is added. The first record consists of the named common block PW. The next records consist of a series of records containing the group dependent P array.

D. Additional Changes to TWOTRAN

For completeness, we list other changes to the TWOTRAN program in Tables XI and XII.

TABLE XI
ADDITIONAL SUBROUTINES

(A) General Subroutines

Name	Remarks
ALDRMP	Control code used to process the additional restart entries in both SCM and LCM required by TWO-TRAN FC and TWOTRAN PW.
ECHECK	Checks a vector for all equal entries.
ECRDMR	Processes vectors from LCM or random disk to dump tape and conversely.
IN	Processes the inward sweeps of INNER.
OUT	Processes the outward sweeps of INNER.
PLOTS	Changed to compute needed values such as flux mesh radii for both the contour and projection plot routines.
PLOTSP	Controls the projection plot code by group.
RDMOW	Reads random disk in parallel with LCM movement to SCM after initially called.
TAKDMR	Reads/writes a vector for a file set.
INCHK	Prepares an indicator table of all zero entries by cross section position from the IDCS table.
IERCK	Checks a vector for all zero entries.

(B) Random Disk Subroutines

FORNW	Converts a unit number to a file set name.
FORGTS	Creates an entry for the file set name in the file set table.
GETIADR	Returns the table address from the file set table for a file set name.
IDONE	Returns the I/O status indicator for a file set name.
RAMPW	Reads/writes random disk for a file set name from LCM.

TABLE XII
ADDITIONAL SYSTEM SUBROUTINES

(A) SC-4020 System Subroutines

Name	Remarks
CONVET	Finds SC-4020 coordinates of a floating point entry.
DLCH	Writes large characters on film.
DRV	Draws a vector between two points on a film.
EXL	Sets lighter exposure than normal on a film.
LOK	Returns normal exposure on film.

(B) Other System Subroutines

MAXM	Finds the maximum of a matrix.
MAXV	Finds the maximum of a vector.
MINM	Finds the minimum of a matrix.
MINV	Finds the minimum of a vector.

Two new common blocks are defined. The contents of the named common block SWEEP are used as a communication link between INER and IN and OUT. The contents of this block are not needed for a restart.

The contents of the named common block CENCOM are used as a communication link between the contour plot routine and other system plot routines.

REFERENCES

1. K. D. Lathrop and F. W. Brinkley, "Theory and Use of the General-Geometry TWOTRAN Program," LA-4432, Los Alamos Scientific Laboratory (1970).
2. A. Salem, "UPDATE - A Program to Maintain a Library of Symbolic Information," LA-1-407A, Los Alamos Scientific Laboratory Computer Library Report (1970).

3. K. D. Lathrop and F. W. Brinkley, "Theory and Use of the General-Geometry TWOTRAN Program," LA-4432, Los Alamos Scientific Laboratory (1970), p. 9.
4. K. D. Lathrop, "Spatial Differencing of the Transport Equations: Positivity vs Accuracy," J. Comput. Phys. 4, 473 (1969).
5. Kaye D. Lathrop, "User's Guide for the TWOTRAN (x,y) Program," LA-4058, Los Alamos Scientific Laboratory (1969).
6. K. D. Lathrop, "Benchmark's Source Situation 3," "Benchmark Problem Solution 5A," and "Benchmark Problem Solution 5B," to be published in the Argonne National Laboratory Benchmark Problem Book, ANL-7416.
7. K. D. Lathrop, "Ray Effects in Discrete Ordinates Equations," Nucl. Sci. Eng. 22, 337 (1966).
8. K. D. Lathrop, "Remedies for Ray Effects," Nucl. Sci. Eng. 45, 235 (1971).
9. Wm. H. Reed, "Spherical Harmonic Solutions of the Neutron Transport Equation from Discrete Ordinate Codes," submitted to Nucl. Sci. Eng.
10. G. I. Bell and S. Glasstone, Nuclear Reactor Theory, (Van Nostrand Reinhold Company, New York, 1970), p. 87.
11. Alvin M. Weinberg and Eugene P. Wigner, The Physical Theory of Neutron Chain Reactions, (University of Chicago Press, Chicago, 1958), p. 266.
12. Bengt G. Carlson, "Simplification of Ray Effect Corrections in Transport Calculations," LA-4757, Los Alamos Scientific Laboratory (1971).
13. Bengt G. Carlson, "Tables of Symmetric Equal Weight Quadrature EQ₂ Over the Unit Sphere," LA-4734, Los Alamos Scientific Laboratory (1971).
14. Jungsung Jang, Hiroshi Chijima, Katsuko Kobayashi, and Hiroshi Nishihara, "Discrete Ordinate Transport Equation Equivalent to P₁ Approximation," submitted to Nucl. Sci. Eng.
15. E. M. Gelbard, personal communication, March 1971.