LA-5428-MS

INFORMAL REPORT

TRIPLET: A Two-Dimensional, Multigroup, Triangular Mesh, Planar Geometry, Explicit Transport Code

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

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

In the interest of prompt distribution, this LAMS report was not edited by the Technical Information staff.

Printed in the United States of America. Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $5.45; Microfiche $1.45

LA-5428-MS
Informal Report
UC-32
ISSUED: October 1973

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

TRIPLET: A Two-Dimensional, Multigroup, Triangular Mesh, Planar Geometry, Explicit Transport Code

by

Wm. H. Reed
T. R. Hill
F. W. Brinkley
K. D. Lathrop

Major support for the computer program was provided by the USAEC Division of Reactor Research and Development.

CONTENTS

FIGURES

TABLES

TRIPLET: A TWO-DIMENSIONAL, MULTIGROUP, TRIANGULAR MESH, PLANAR GEOMETRY, EXPLICIT TRANSPORT CODE

by

Wm. H. Reed

T. R. Hill

F. W. Brinkley

K. D. Lathrop

ABSTRACT

1. Program Identification: TRIPLET (LP-0142)
2. Computer for which program is designed and others on which it is operational: CDC-7600, CDC-6600, IBM-360/195.
3. Description of Function: TRIPLET solves the two-dimensional multigroup transport equation in planar geometries using a regular triangular mesh. Regular and adjoint, inhomogeneous and homogeneous (k_{eff} and eigenvalue searches) problems subject to vacuum, reflective or source boundary conditions are solved. General anisotropic scattering is allowed and anisotropic distributed sources are permitted.
4. Method of Solution: The discrete ordinates approximation is used for the angular variables. A finite element method in which the angular flux is assumed to be given by a low-order polynomial in each triangle is used to solve the discrete ordinates equations. Angular fluxes are allowed to be discontinuous across triangle boundaries, and the order of the polynomial is input data to the code. Both inner (within-group) and outer iteration cycles are accelerated by either system or fine mesh rebalance.
5. Restrictions: Variable dimensioning is used so that any combination of problem parameters leading to a container array less than MAXLEN can be accommodated. On CDC machines MAXLEN can be about 40,000 words and peripheral storage is used for most group-dependent data. On IBM machines TRIPLET will execute in single precision (4 bytes per word) so that MAXLEN can be several hundred thousand and most problems can be core contained.
6. Running Time: A six group, S_2, 1700 triangle, k_{eff} calculation of an EBR-II core requires about 4.4 minutes of CDC-7600 time. Running times vary almost linearly with the total number of unknowns.
7. Unusual Features of the Program: Sources, fluxes, S_n constants, and cross sections may be input from standard interface files. Creation of standard interface output files for S_n constants and scalar and angular fluxes is optional. All binary data transfers are localized in subroutines called REED and RITE. Flexible edit options, including dumps and restart capability, are provided.
8. Machine Requirements: Six output (scratch) units, five interface units (use of interface units is optional), and two system input/output units are required. A large bulk memory is necessary if core storage is inadequate, as on the CDC machines.
9. Related Programs: TRIPLET is based in large part on the two-dimensional orthogonal mesh code TWOTRAN-II, which is an improved version of the TWOTRAN program.
10. Material Available: Source deck, test problems, results of executed test problems, and this report are available from the Argonne Code Center.

I. INTRODUCTION

TRIPLET is a program for solving the two-dimensional neutron transport equation in planar geometries using a triangular spatial mesh. Planar geometries are infinite in extent in one dimension with material properties depending only on the other two dimensions. In other words, every planar slice through the system perpendicular to the dimension of infinite extent is identical. Examples of such geometries are the (x,y) and (r,θ) orthogonal geometries.

The advantages of a triangular mesh in such geometries are several. Systems of physical interest are often of complicated, curved design. While the description of such systems with an orthogonal mesh is always possible, these shapes are often described to greater accuracy by a triangular mesh with many fewer mesh cells. Hexagonal lattice geometries are often of interest, and such lattices can be described exactly by a triangular grid. Nonrectangular system boundaries can be treated with triangles while preserving convexity, often with a large saving in number of mesh cells.

The increased flexibility of a triangular mesh is not without added cost. Description of the mesh itself is difficult, and users of the TRIPLET program must expect that preparation of the input data for this code will be more difficult than for the orthogonal grid codes. Complexities in the TRIPLET solution algorithms caused by the triangular mesh lead to execution times that are about twice as long as those for the orthogonal mesh code TWOTRAN,[1] on a per mesh cell basis. Thus it is clear that TRIPLET is complementary to, not a replacement for, the TWOTRAN code and should be used only when an orthogonal mesh is incapable of giving accurate answers efficiently.

The regular triangular mesh used by TRIPLET is almost as flexible as a general triangular mesh but is much easier to describe. A regular triangular mesh is defined by the following properties:

1. Adjacent triangles must share two common vertices.

2. All triangles must lie on horizontal bands extending across the system. Equivalently, all triangle vertices must lie on horizontal lines extending across the system.

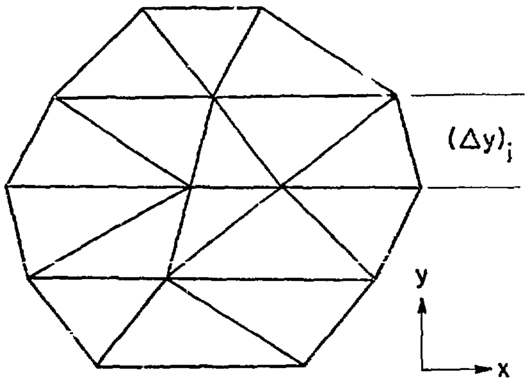


Fig. 1. A typical regular triangular mesh.

3. Each interior vertex must be common to six adjacent triangles.

Note that we do not require that triangles be equilateral or outer boundaries be rectangular. An example of a regular triangle mesh is illustrated in Fig. 1.

There are two reasons why TRIPLET uses a regular mesh. First, specification of a regular triangular mesh is much simpler than specification of a general triangular mesh. Only four pieces of data need be input to the code: the number of triangles on each band, the band widths $(\Delta y)_j$ (see Fig. 1), the x coordinates of the vertices along each horizontal line, and the orientation of the first triangle on each band. This orientation is specified by indicating whether the triangle points up or down. A triangle is said to be upward pointing if it has two vertices on the lower line and one on the upper line forming the band, and vice versa. A regular triangular mesh is determined uniquely by the above data.

The second reason a regular mesh is used by TRIPLET is related to our decision to use explicit methods for solving the discrete ordinates equations. Explicit methods necessarily sweep the mesh in the direction of the characteristics (the direction of neutron travel), so that there is a definite order in which the triangles must be solved. This order depends upon the direction of neutron flow across triangle boundaries and is not straightforward as for an orthogonal mesh. Thus the direction of flow across boundaries must be determined and decisions made as to how to progress through the mesh. These decisions are much more complicated and time-consuming for a general triangular mesh than for a regular triangular mesh. Since the order in which the mesh is swept differs with each discrete ordinate direction, these decisions must be made repetitively in the innermost iteration loops of TRIPLET. If complicated, such decisions would be prohibitively expensive. We restrict to a regular mesh to simplify these decisions as much as possible while retaining most of the flexibility of a general mesh.

In addition to its triangular mesh geometry, TRIPLET is unique in its use of explicit finite element methods for solving the discrete ordinates equations. These methods assume a piecewise polynomial representation for the angular flux in each direction. Specifically, the angular flux is assumed to be given by a low-order polynomial within each triangle; the order of the polynomial is input data to the code. The angular flux is allowed to be discontinuous across triangle boundaries and is determined within each triangle by requiring that the residual be orthogonal to the low-order polynomials. This procedure has been found to yield a very accurate and stable method.

Most other features and options of the TRIPLET program are similar to those of TWOTRAN. TRIPLET was written by modifying TWOTRAN, and the codes share the same overlay structure and have many common subroutines. The major features of TRIPLET include:

1. direct or adjoint capability,
2. anisotropic scattering of any order,
3. built-in S_n constants (nonsymmetric quadrature sets can be input if desired),
4. vacuum, reflective, or source boundaries on all sides (a special boundary condition, useful for hexagonal cell calculations, is allowed on the right-hand boundary),
5. nonrectangular domains,
6. source, k_{eff}, alpha, or concentration search problems,
7. whole system or fine mesh rebalance convergence acceleration for inner and outer iterations,
8. optional print suppress for large input arrays,
9. space dependent buckling height,
10. flexible edit and restart options,
11. optional input of flux, source, S_n constants, and cross sections from standard interface files,
12. optional output of scalar and angular fluxes and S_n constants in standard interface format.

Because TRIPLET is similar to the TWOTRAN code from which it was derived, much of the TWOTRAN manual[1] applies directly to TRIPLET. To obtain a unified and self-contained presentation for the convenience of the reader of this report, we have reproduced many portions of the TWOTRAN manual in what follows. The next section of this report contains the theoretical development of all the methods and approximations used in TRIPLET. Section III is a users guide and Sec. IV contains detailed programming information. The contents of this report follow the guidelines[2] for documentation of digital computer programs proposed as an American Nuclear Society standard.

II. THEORY

In this section the energy, angular, and spatial variables of the transport equation are discretized to obtain a set of linear algebraic equations. The exact transport equation is written and discussed in Sec. II.A. The spherical harmonic expansion of scattering sources, the multigroup treatment of the energy variable, and the discrete ordinates approximation for the angular variables are all treated in Sec. II.B. Section II.C is devoted entirely to a discussion of the finite element methods used to solve the discrete ordinates equations on a triangular mesh. The solution of the linear algebraic system obtained by these approximations is discussed in Sec. II.D.

A. The Analytic Transport Equation

The time-independent, inhomogeneous transport equation is written

$$\nabla \cdot (\underline{\Omega}\Psi) + \sigma_t\Psi(\underline{r},E,\underline{\Omega}) =$$

$$\iint dE'd\Omega' \; \Psi(\underline{r},E',\Omega')\sigma_s(E' \rightarrow E,\underline{\Omega}' \cdot \underline{\Omega}) \qquad (1)$$

$$+ \chi(E) \iint dE'd\Omega' \; \Psi(\underline{r},E',\underline{\Omega}')\nu\sigma_f(E')/4\pi + Q(\underline{r},E,\underline{\Omega}),$$

in which Ψ is the particle flux (number density of particles times their speed) defined such that

$\Psi dVdEd\Omega$ is the flux of particles in the volume element dV about $\underline{r}$, in the element of solid angle $d\Omega$ about $\underline{\Omega}$, in the energy range dE about E. Similarly, $QdVdEd\Omega$ is the number of particles in the same element of phase space emitted by sources independent of Ψ. The macroscopic total interaction cross section is denoted by σ_t, the macroscopic scattering transfer probability (from energy E' to E through an angle with cosine $\underline{\Omega}'\cdot\underline{\Omega}$) by σ_s, and the macroscopic fission cross section by σ_f. All of these quantities may depend on $\underline{r}$, but we have omitted this argument for simplicity. The number of particles emitted isotropically ($1/4\pi$) per fission is ν, and the fraction of these liberated in the range dE about E is $\chi(E)$. This fraction may actually depend on both $\underline{r}$ and the fissioning species, but such possibilities are not admitted in the TRIPLET program.

The homogeneous transport equation is written in the same manner as Eq. (1) except that Q is zero and the term representing a source of neutrons due to fission is divided by the eigenvalue k_{eff}. In this report the inhomogeneous problem will be referred to as a source problem and the homogeneous or eigenvalue problem will be referred to as a k_{eff} problem. The TRIPLET program will solve both types of problems.

Throughout this report we will use the (x,y) form of Eq. (1). Here the operator $\nabla\cdot\underline{\Omega}$ is given by

$$\nabla\cdot\underline{\Omega}\Psi = \mu\frac{\partial\Psi}{\partial x} + \eta\frac{\partial\Psi}{\partial y} \quad , \qquad (2)$$

where μ and η are the direction cosines of $\underline{\Omega}$ with respect to the x and y axes, thus

$$\mu = \hat{e}_x\cdot\underline{\Omega}$$

$$\eta = \hat{e}_y\cdot\underline{\Omega} \quad .$$

B. Energy and Angular Approximations

In this section the spherical harmonic expansion of the scattering sources is presented, and the multigroup discrete ordinates equations are derived.

1. Spherical Harmonic Expansion of Source Terms

In the TRIPLET program, the scattering transfer probability is assumed to be represented by a finite Legendre polynomial expansion:

$$\sigma_s(E'\to E,\mu_o) = \sum_{n=0}^{ISCT}\frac{2n+1}{4\pi}P_n(\mu_o)\sigma_{sn}(E'\to E) \quad ,$$

where $\mu_o = \underline{\Omega}\cdot\underline{\Omega}' = \mu\mu' + (1-\mu^2)^{\frac{1}{2}}(1-\mu'^2)^{\frac{1}{2}}\cos(\phi-\phi')$. If this expansion is inserted in Eq. (1), and if the addition theorem is used to expand $P_n(\mu_o)$, we can write

$$\int\int d\Omega'\Psi(\underline{r},E',\underline{\Omega}')\sigma_s(E'\to E,\mu_o) = \sum_{n=0}^{ISCT}\frac{2n+1}{2\pi}\sigma_{sn}(E'\to E)$$

$$\cdot\sum_{k=0}^{n}R_n^k(\mu,\phi)\int_{-1}^{1}d\mu'\int_0^{\pi}d\phi' R_n^k(\mu',\phi')\Psi \quad . \qquad (3)$$

In deriving this expression, the ϕ symmetry of Ψ is used, reducing the domain of $\underline{\Omega}$ to a hemisphere and eliminating expansion terms odd in ϕ. The functions R_n^k are defined by

$$R_n^k = \left[\frac{(2-\delta_{ko})(n-k)!}{(n+k)!}\right]^{\frac{1}{2}} P_n^k(\mu)\cos k\phi \quad , \qquad (4)$$

where δ_{ko} is the Kronecker delta (equal to 1 when k = 0, and vanishing otherwise) and the R_n^k are the associated Legendre polynomials. These functions are orthogonal; i.e.,

$$\int_{-1}^{1}d\mu\int_0^{\pi}d\phi R_n^k(\mu,\phi)R_m^\ell(\mu,\phi) = \frac{2\pi}{2n+1}\delta_{nm}\delta_{k\ell} \quad . \qquad (5)$$

Hence, if the angular flux is expanded in a series of these functions,

$$\Psi = \sum_{n=0}^{\infty}(2n+1)\sum_{k=0}^{n}R_n^k\Psi_n^k \quad , \qquad (6)$$

the expansion coefficients are given by

$$\Psi_n^k = \int_{-1}^{1}d\mu\int_0^{\pi}d\phi R_n^k\Psi/2\pi \quad . \qquad (7)$$

Using this formula, we can rewrite Eq. (1) as

$$\nabla\cdot(\underline{\Omega}\Psi) + \sigma_t\Psi =$$

$$\int_0^\infty dE' \sum_{n=0}^{ISCT} (2n+1)\sigma_{sn}(E' \rightarrow E) \sum_{k=0}^{n} R_n^k(\mu,\phi)\Psi_n^k$$

$$+ \chi(E) \int_0^\infty dE' \; \nu\sigma_f(E')\Psi_o^o + \sum_{n=0}^{IQAN} (2n+1) \sum_{k=0}^{n} R_n^k Q_n^k \quad . \tag{8}$$

As implied by the sum in this equation, we have assumed that the source Q is represented by IQAN+1 terms of an expansion like Eq. (6).

2. Multigroup Equations

The energy domain of interest is assumed to be partitioned into IGM intervals of width ΔE_g, g = 1, 2, ..., IGM. By convention, increasing g represents decreasing energy. If we integrate Eq. (8) over ΔE_g we can write

$$\nabla\cdot(\underline{\Omega}\Psi_g) + \sigma_{tg}\Psi_g =$$

$$\sum_{h=1}^{IGM} \sum_{n=0}^{ISCT} (2n+1)\, \sigma_{snh\rightarrow g} \sum_{k=0}^{n} R_n^k \Psi_{nh}^k$$

$$+ \chi_g \sum_{h=1}^{IGM} \nu\sigma_{fh}\, \Psi_{oh}^o + \sum_{n=0}^{IQAN} (2n+1) \sum_{k=0}^{n} R_n^k Q_{ng}^k \quad . \tag{9}$$

Here, the flux for group g,

$$\Psi_g = \int_{\Delta E_g} \Psi dE \quad , \tag{10}$$

is no longer a distribution in energy, but is the total number of particles in the energy interval. For this reason, when group structures are changed, the effect on results must be evaluated by comparing $\Psi_g/\Delta E_g$. Because of Eq. (10), energy integrals in TRIPLET are evaluated by simple sums.

The cross sections subscripted with g are averages, e.g.,

$$\sigma_{tg} = \int_{\Delta E_g} \sigma_t \Psi dE \Bigg/ \int_{\Delta E_g} \Psi dE \quad , \tag{11}$$

but, of course, Ψ is not known and must be approximated by some means. If in Eq. (11) the angular dependence of Ψ is nonseparable, then σ_{tg} will depend on angle. No provision for such dependence is made in TRIPLET. Recipes for taking this dependence into account, as well as for improving the averages $\sigma_{snh\rightarrow g}$ when scattering is severely anisotropic, are given by Bell, Hansen, and Sandmeier.[3]

3. Discrete Ordinates Equations

The three terms on the right-hand side of Eq. (9) represent sources of neutrons due to scattering, fission, and inhomogeneous sources, respectively. All coupling between the IGM multigroup equations is contained in these terms. To simplify notation for the following analysis, we write all three sources simply as S_g, which we will treat as a known quantity. Of course, S_g depends on the unknown flux Ψ_g, but we will generate S_g iteratively using latest values of Ψ_g. For details of the iterative processes in TRIPLET see Sec. II.D.4. Dropping the group subscript g and using the form of the divergence operator given in Eq. (2), we have

$$\mu \frac{\partial\Psi}{\partial x} + \eta \frac{\partial\Psi}{\partial y} + \sigma_t\Psi = S \quad , \tag{12}$$

where the functions Ψ and S depend upon the variables x, y, μ, and η. We will now be concerned with the discretization of Eq. (12) with respect to the variables μ and η.

We first assume that a set of MM quadrature points (μ_m,η_m) has been selected on the unit disk $(\mu^2 + \eta^2 \leq 1)$, along with a set of quadrature weights w_m normalized so that $\sum_{m=1}^{MM} w_m = 1$. The user of TRIPLET may input this quadrature to the code or may use standard built-in sets, if desired. Although the built-in quadrature sets are symmetric under 90° rotations, for vacuum boundaries there are no symmetry requirements on input quadratures, and MM may be any positive integer greater than or equal one. If built-in quadratures are used, MM must equal one of a selected set of values for which quadratures are available (see Sec. III.B.4). If reflecting or special boundaries are used, then input quadratures must meet certain symmetry requirements which are also discussed in Sec. III.B.4.

The source S that is the right-hand side of Eq. (9) involves moments Ψ_n^k of the angular flux Ψ. With the aid of the above quadrature, these moments can be approximated as

$$\Psi_n^k(x,y) \approx \sum_{m=1}^{MM} w_m R_n^k(\mu_m,\phi_m)\Psi(x,y,\mu_m,\eta_m) \quad , \qquad (13)$$

where ϕ_m is defined in TRIPLET by

$$\phi_m = \tan^{-1}(1 - \mu_m^2 - \eta_m^2)^{\frac{1}{2}}\ \eta_m \qquad , \eta_m > 0 \quad ,$$

$$= \tan^{-1}(1 - \mu_m^2 - \eta_m^2)^{\frac{1}{2}}\ \eta_m + \pi \ , \ \eta_m < 0 \quad . \qquad (14)$$

With the definition of an approximate angular flux in the m'th direction as

$$\Psi_m(x,y) \approx \Psi(x,y,\mu_m,\eta_m) \quad ,$$

we may write the following set of equations as an approximation for Eq. (12):

$$\mu_m \frac{\partial\Psi_m}{\partial x} + \eta_m \frac{\partial\Psi_m}{\partial y} + \sigma_t\Psi_m = S_m,\ m = 1,\ 2,\ \dots\ MM, \qquad (15)$$

where the angular source S_m in the m'th direction is computed from

$$S_m = \sum_{h=1}^{IGM} \sum_{n=0}^{ISCT} (2n+1)\ \sigma_{snh\to g} \sum_{k=0}^{n} R_n^k(\mu_m,\phi_m)\tilde{\Psi}_{nh}^k$$

$$+ \chi_g \sum_{h=1}^{IGM} \nu\sigma_{fh}\ \tilde{\Psi}_{oh}^o + \sum_{n=0}^{IQAN} (2n + 1)$$

$$\cdot \sum_{k=0}^{n} R_n^k(\mu_m,\phi_m)Q_{ng}^k \quad . \qquad (16)$$

The approximate moments $\tilde{\Psi}_{nh}^k$ appearing above are computed from the approximate angular flux in group h as

$$\tilde{\Psi}_{nh}^k(x,y) = \sum_{m=1}^{MM} w_m R_n^k(\mu_m,\phi_m)\Psi_m^h(x,y) \quad . \qquad (17)$$

Eqs. (15) are known as the discrete ordinates equations, and these equations represent a coupled set of first-order differential equations for the approximate angular flux $\Psi_m(x,y)$. All coupling in this set of equations, both energy and angle, is contained in the source term S_m.

C. Discretization of Spatial Variables

The approximations that have been made so far in this report are independent of the form of the spatial mesh used by a transport code. These approximations are in fact identical to those used in the orthogonal grid code TWOTRAN. In this section, however, we will depart from the standard diamond difference scheme used in that code and will develop a set of finite element methods for spatial discretization of the discrete ordinates equations on a triangular mesh.

The diamond difference scheme is an example of an explicit difference scheme. Difference schemes for the transport equation fall into two broad categories, which we will refer to as implicit and explicit methods. In an implicit method no attempt is made to solve in the direction in which neutrons are streaming. Instead, variational or Galerkin methods are used to determine a set of linear algebraic equations for all the unknowns. This set of equations is then solved, usually by direct methods, to obtain the final solution. An explicit method, on the other hand, sweeps once through the mesh, solving for the unknowns in the direction in which neutrons are streaming. Of course, this is also equivalent to solving a set of linear algebraic equations, but here the matrix to be inverted is triangular, or at least block triangular. Perhaps the clearest distinction between the two types of methods can be made in the following way. In an explicit method a particular mesh cell is coupled only to those mesh cells visible when looking backward along the direction in which neutrons are traveling. This direction is determined for each discrete ordinates equation by the direction cosines μ_m and η_m appearing in Eq. (15). An implicit method couples all adjacent mesh cells with no regard for the direction of neutron travel. The finite element methods developed below, like the diamond difference scheme, are explicit in nature.

Our task is now the development of a discrete (in x and y) approximation to Eq. (15) on a triangular mesh. We note again that the MM discrete ordinates equations are coupled only through the source term S_m, which is computed in an iterative context using previous values of the angular fluxes. Hence, for the sake of the present analysis, we may treat S_m as a known quantity. Let us fix our attention upon a particular discrete ordinates equation, which we write without the angular subscript m as

$$\mu \frac{\partial\Psi}{\partial x} (x,y) + \eta \frac{\partial\Psi}{\partial y} (x,y) + \sigma_t(x,y)\Psi(x,y) = S(x,y) \tag{18}$$

Finite element methods for solving partial differential equations like Eq. (18) usually involve the assumption that the unknown function, in this case Ψ, can be approximated by some member of a finite-dimensional set of functions. This set of functions is often referred to as the trial space. A particular member of this space of functions is selected by some procedure like minimizing a functional or requiring the residual to be orthogonal to a set of weighting functions. The selected member is the desired approximate solution to the partial differential equation.

The finite element methods used in TRIPLET are derived as indicated above using a weight and integrate technique. The trial space consists of functions that are piecewise polynomial and discontinuous across triangle boundaries. More precisely, if $\tilde{\Psi}_i(x,y)$ is our approximation for the exact solution $\Psi(x,y)$ within the i'th triangle, then we assume that

$$\tilde{\Psi}_i(x,y) = \sum_{j=0}^{N} \sum_{k=0}^{N-j} A_{jk}^{(i)} x^j y^k \quad , \tag{19}$$

where N is the polynomial order and the coefficients $A_{jk}^{(i)}$ represent unknowns to be determined by a weight and integrate procedure. To complete the specification of our trial space, we must assign a unique value to the approximate flux on the triangle boundaries. It is essential to the following analysis that we assume that the flux on a triangle boundary is the limit of the flux as one approaches the boundary in the direction in which neutrons are streaming.

There are many ways in which a polynomial in x and y can be expressed. The representation of Eq. (19) is certainly the most common, but it is inconvenient for our purposes because the coefficients $A_{jk}^{(i)}$ have little physical meaning. We prefer, instead, to use a Lagrange representation of the polynomials with which we work. Let us assume that a set of $K \equiv (N+1)(N+2)/2$ distinct points (x_k,y_k) have been placed on the triangle of interest, where N is the order of the polynomial to be represented. The placement of these points is discussed below. We use K points because there are K linearly independent polynomials of order less than or equal N. We define the polynomial $L_i^{(k)}(x,y)$ as the unique polynomial of order less than or equal N that is unity at the point (x_k,y_k) and is zero at the other K-1 points on the i'th triangle. We refer to the K polynomials as Lagrange polynomials. If the points (x_k,y_k) have been chosen properly, then the Lagrange polynomials exist, are linearly independent, and form a basis for the space of polynomials of order less than or equal N. Thus we can replace Eq. (19) by the following equation with no loss of content:

$$\tilde{\Psi}_i(x,y) = \sum_{k=1}^{K} \tilde{\Psi}_i^{(k)} L_i^{(k)}(x,y) \quad . \tag{20}$$

The unknown coefficients $\tilde{\Psi}_i^{(k)}$, which have replaced the unknowns $A_{jk}^{(i)}$ of Eq. (19), can be interpreted as the values of the approximate solution $\tilde{\Psi}_i(x,y)$ at the points (x_k,y_k) in the i'th triangle. It is this physical interpretation for $\tilde{\Psi}_i^{(k)}$ which leads us to the Lagrange representation for $\tilde{\Psi}_i(x,y)$.

There are many arrangements of K points on a triangle that will guarantee uniqueness and linear independence of the Lagrange polynomials. We choose a particular placement of these points which makes the treatment of the triangle boundaries simple. For an N'th order polynomial, we place N+1 equally spaced points on each face of the triangle, with a point at each vertex. The remaining points are distributed uniformly within the interior of the triangle. Fig. 2 illustrates the placement of these points for a few low-order polynomials. Because a polynomial in x and y of order N is uniquely determined on a straight line by values at N+1 distinct points on the line, the flux along the triangle boundary is determined by the N+1 points on that boundary, without regard

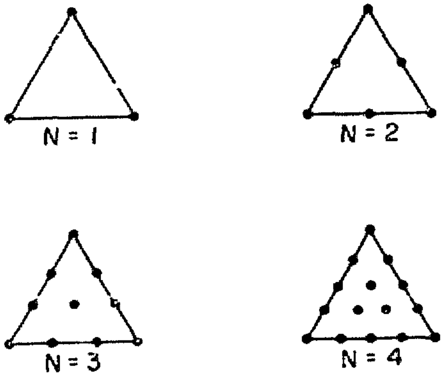


Fig. 2. The triangular point arrangement for a few low-order polynomials.

for the other points on the triangle. The boundary flux is in fact given by the unique N'th order one-dimensional polynomial that assumes given values at these N+1 points.

With the above definition of the trial space in which we look for an approximate solution, there are KI unknowns to be determined, where K is the number of unknowns per triangle and I is the total number of triangles. The minimum polynomial order accepted by TRIPLET is N=1, in which case K=3 and there are a total of 3I unknowns. Our next task is to derive a set of KI equations for these KI unknowns. Because we are only interested in explicit methods, this task reduces to the task of deriving a set of K equations for the K unknowns on a given triangle, assuming that the flux on incoming boundaries is known from boundary conditions or from prior calculation in adjoining cells. An incoming boundary is a triangle boundary across which the neutron flow is directed into the cell. Of course, an incoming boundary for one cell is an outgoing boundary for an adjacent cell, and the definition of an incoming boundary depends upon the direction (μ_m, η_m) under consideration. There are two cases that must be considered: one or two faces may be visible when looking along the direction $\underline{\Omega}_m$ determined by μ_m and η_m, hence there may be one or two incoming boundaries. These two cases are depicted in Fig. 3.

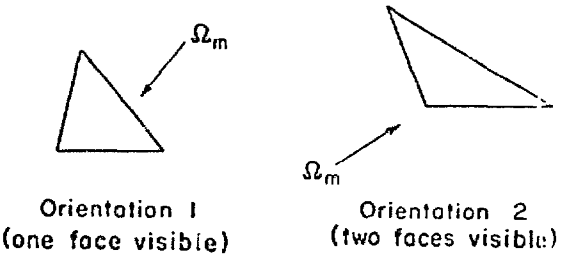


Fig. 3. The two possible orientations of a triangle with respect to a direction $\underline{\Omega}_m$.

The form of the Lagrange polynomials $L_i^{(k)}(x,y)$ of Eq. (20) depends upon the origin of the (x,y) coordinate system and the shape of the triangle under consideration. We therefore find it convenient, when solving for the unknown fluxes $\tilde{\psi}_i^{(k)}$ on a particular triangle, to map every triangle to a standard form. Such a mapping can be accomplished by a linear change of independent variables from x and y to x' and y'. Thus we assume that

$$
\begin{aligned}
x' &= ax + by + c \\
y' &= dx + ey + f \quad ,
\end{aligned}
\tag{21}
$$

where the coefficients a through f are chosen so that every point (x,y) on the triangle of interest is mapped in a one-to-one fashion onto a point (x', y') of a right triangle with legs of unit length. The x'-y' coordinate system is illustrated in Fig. 4. The transformation of Eq. (21) is not unique, but we do observe the additional convention that the hypotenuse represents the incoming boundary for triangles of orientation 1 and the legs represent the incoming boundaries for triangles of orientation 2.

The change of variables of Eq. (21) gives a transport equation with altered direction cosines. We have

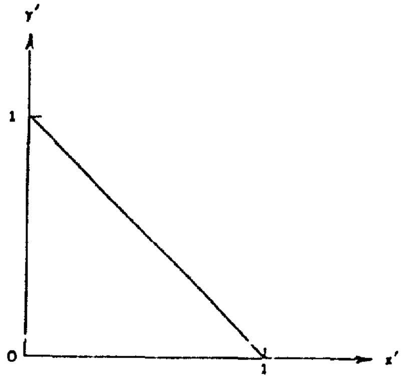


Fig. 4. The standard right triangle and the x'-y' coordinate system.

$$\frac{\partial \Psi}{\partial x} = a \frac{\partial \Psi}{\partial x'} + d \frac{\partial \Psi}{\partial y'}$$

$$\frac{\partial \Psi}{\partial y} = b \frac{\partial \Psi}{\partial x'} + e \frac{\partial \Psi}{\partial y'} \quad ,$$

so that

$$\mu' \frac{\partial \Psi}{\partial x'} + \eta' \frac{\partial \Psi}{\partial y'} + \sigma \Psi = S \quad , \tag{22}$$

where

$$\mu' = a\mu + b\eta$$
$$\eta' = d\mu + e\eta \quad . \tag{23}$$

This change of variables is therefore easily accomplished by a simple change of μ and η as indicated above.

We may now solve Eq. (22) for the unknown fluxes $\tilde{\Psi}_i^{(k)}$ on the right triangle of Fig. 4, instead of solving Eq. (18) for these same fluxes on the original triangle. We again assume that

$$\tilde{\Psi}_i(x',y') = \sum_{k=1}^{K} \tilde{\Psi}_i^{(k)} L^{(k)}(x',y') \quad , \tag{24}$$

where the Lagrange polynomials are defined on the standard right triangle and are independent of the subscript i.

We will now derive a set of K equations for our K unknowns. Let us assume that we are dealing with a triangle of orientation 2, so that the legs of the right triangle of Fig. 4 represent incoming boundaries. Therefore, the flux on these boundaries is known from boundary conditions or from prior calculation in adjoining cells, and the function $\tilde{\Psi}_i(x',y')$ of Eq. (24) represents the flux in the interior of the triangle and on the hypotenuse by continuity. The flux may be discontinuous along $x'=0$ and along $y'=0$. We assume the flux on the bottom boundary is given by $f(x')$ and the flux on the left boundary is $g(y')$. Substitution of this discontinuous flux into the transport Eq. (22) gives

$$\mu' \sum_{k=1}^{K} \left[\tilde{\Psi}_i^{(k)} L^{(k)}(0,y') - g(y') \right] \delta(x')$$

$$+ \eta' \sum_{k=1}^{K} \left[\tilde{\Psi}_i^{(k)} L^{(k)}(x',0) - f(x') \right] \delta(y')$$

$$+ \sum_{k=1}^{K} \left[\mu' \frac{\partial L^{(k)}}{\partial x'}(x',y') + \eta' \frac{\partial L^{(k)}}{\partial y'}(x',y') \right.$$

$$\left. + \sigma_t L^{(k)}(x',y') \right] \tilde{\Psi}_i^{(k)} - S(x',y')$$

$$\equiv R(x',y') \quad . \tag{25}$$

where $R(x',y')$ are referred to as the residual. The two terms containing the Dirac delta functions $\delta(x')$ and $\delta(y')$ are due to the boundary discontinuities. It should be noted that the function $L^{(k)}(0,y')$ is identically zero unless the k'th point is located on the left boundary and the function $L^{(k)}(x',0)$ is identically zero unless the k'th point is located on the bottom boundary.

We observe that, in general, the residual R cannot identically equal zero for any choice of $\tilde{\Psi}_i^{(k)}$, since the exact solution is rarely given by a low-order polynomial. We introduce a set of K linearly independent weight functions $W_j(x',y')$, $j = 1, 2, \ldots K$, and insist that

$$(W_j, R) = 0, \; j = 1, 2, \ldots K \quad , \tag{26}$$

where (a,b) represents an integration of ab over the right triangle. Equation (26) represents a K by K linear algebraic system of equations for the unknowns $\tilde{\Psi}_i^{(k)}$. With a proper choice of weight functions, this system is nonsingular and can be solved by any of a variety of methods appropriate for small linear systems. In TRIPLET, the weight functions W_j are chosen to be the low-order polynomials of the form $(x')^m(y')^n$ which span the space of polynomials of order less than or equal N, and Eq. (26) is solved by Gaussian elimination with partial pivoting.

For triangles of orientation 1, the linear system solved by TRIPLET to obtain the K unknowns $\tilde{\Psi}_i^{(k)}$ is the same as Eq. (26) above, except for boundary terms. In this case, the hypotenuse is an incoming boundary and the legs are outgoing boundaries, so that the flux discontinuity occurs along the line $x' + y' = 1$. We assume that the incoming boundary flux is given by $f(z)$, where $z = 0$ at the lower right-hand corner of the triangle and $z = \sqrt{2}$ at the upper corner of the triangle. The residual is now given by

$$R(x',y') = \frac{\mu' + \eta'}{\sqrt{2}} \left| f\left(\left[(1 - x')^2 + (y')^2 \right]^{\frac{1}{2}} \right) - \sum_{k=1}^{K} \tilde{\psi}_i^{(k)} L^{(k)}(x',y') \right| \cdot \delta(x + y - 1) + \sum_{k=1}^{K} \left| \mu' \frac{\partial L^{(k)}}{\partial x'}(x',y') + \eta' \frac{\partial L^{(k)}}{\partial y'}(x',y') + \sigma_t L^{(k)}(x',y') \right| \tilde{\psi}_i^{(k)} - S(x',y') \ . \quad (27)$$

The weight functions are again chosen to be low-order polynomials of the form $(x')^m (y')^n$ spanning the space of polynomials of order less than or equal N, and the unknowns $\psi_i^{(k)}$ are determined by solving the K by K linear system of the form of Eq. (26).

Numerical experiments[10] have shown that the finite element methods described above are capable of yielding accurate solutions of the discrete ordinates equations. These methods are theoretically capable of yielding high order (greater than second order) accuracy for $N > 1$, but in practice the angular flux is not sufficiently smooth for this to happen. First derivatives of the angular flux almost always fail to exist along certain characteristics, so that all methods are first-order accurate in predicting local fluxes. We find experimentally that eigenvalues and region-averaged quantities are predicted to second-order accuracy by all these methods. Because of this, TRIPLET is usually most efficient when run with $N = 1$.

One disadvantage of these methods that are based on discontinuous representations of the angular flux is the large amount of core storage required. It is necessary to store the scalar flux and enough moments of the angular flux to generate the scattering source at all KI points on the spatial grid. Even for $N = 1$ we have $K = 3$, and 3I words of storage are required for each moment of the angular flux. Orthogonal mesh codes using the diamond difference scheme need only I words of storage for each moment, where I is the total number of spatial mesh cells. Such a large increase in storage requirements can mean that some large problems will not fit in available core.

To minimize core storage requirements, an option is allowed in TRIPLET so that the user may select a less elegant and less accurate solution method. This solution method assumes that the angular flux is given within each triangle by a linear polynomial and imposes sufficient flux continuity across triangle boundaries to eliminate two of the three unknowns in each triangle. The equation solved for the single remaining unknown is always the balance equation obtained by integrating Eq. (22) over the triangle. For triangles of orientation 1 the procedure is straightforward. We simply impose continuity across the single incoming boundary, so that the two fluxes on that boundary are determined. The third flux is obtained by solving the balance equation. Strict continuity cannot be imposed for triangles of orientation 2, however, since that would fully determine all three points within the triangle, and neutron balance could not be guaranteed. For triangles of orientation 2 we follow this prescription. We first impose continuity across one of the two incoming boundaries and solve the balance equation for the remaining flux. We then impose continuity across the other boundary and again solve the balance equation for the third flux. The final solution is then obtained by averaging the two fluxes obtained at each of the three points. In every case we solve only the balance equation, which in turn involves only cell-average sources. For this reason, only cell-average fluxes need be stored to generate these sources. Therefore, core storage requirements for flux and source arrays are reduced by the factor 1/3 when TRIPLET is run with this continuous method.

The continuous method is almost as accurate as the discontinuous method but a great deal less stable. Negative flux values can be produced by either method, but negative fluxes are encountered much more often with the continuous method. There is no negative flux fixup in TRIPLET, and if the user is unhappy with such fluxes he must switch to the discontinuous method, use a finer mesh, or both.

D. Solution Algorithms

1. Boundary Conditions

The equations and methods of Sec. II.C determine the angular flux within a triangle provided the incoming flux on triangle boundaries is known. This incoming flux may be known from prior calculation in adjacent cells. If an incoming triangle boundary is a system boundary, then the incoming flux must be determined from the appropriate boundary condition. In TRIPLET, one of the following boundary conditions must be applied to the top, bottom, and each of the two sides of a system. The top and bottom boundaries are defined to be those parts of the system perimeter coinciding with the top and bottom horizontal lines (node lines), respectively. Note that the top and bottom boundaries are always flat, but side boundaries may approximate some curved surface. It is possible for the top or bottom boundary, or both, to fail to exist. This happens, for example, if there is a single upward pointing triangle on the top band. In these cases the boundary conditions are meaningless and are not used.

a. Vacuum Boundary Condition

The value of the angular flux on the boundary is set equal to zero for all incoming directions.

b. Reflective Boundary Condition

Reflective boundary conditions may be applied to each of the four sides in any combination. TRIPLET users are cautioned, however, to note that, if the external boundary is not rectangular, the use of reflective boundaries may lead to a nonconvex domain of interest in which reentrant neutrons are not properly accounted for. Furthermore, a reflective boundary must be parallel with one of the coordinate axes, and the quadrature must possess certain symmetries (see Sec. III.B.4). If either of the side boundaries is reflective, the quadrature must be symmetric about the line $\eta = 0$, whereas if either the top or bottom boundary is reflective, the quadrature must be symmetric about $\mu = 0$. These requirements are necessary to guarantee that all reflections of directions are contained in the quadrature set.

For a reflective boundary condition, the value of the incoming flux on the boundary is set equal to the value of the outgoing flux at the same spatial position in the direction corresponding to specular reflection. At the right-hand boundary, for example, we set

$$\Psi_{\text{incoming}}(-\mu,\eta) = \Psi_{\text{outgoing}}(\mu,\eta) \quad ,$$

$$\mu > 0 \quad .$$

c. Special Boundary Condition

A special boundary condition useful for hexagonal cell calculations is provided in TRIPLET. This boundary condition can be used on the right-hand boundary only and must be used in conjunction with reflective boundary conditions on the other three sides. When a special boundary condition is used, the band widths must be symmetric about a horizontal line through the middle of the system. There may be an even or odd number of bands; if there is an odd number of bands the centermost band is unpaired and may be any size.

Figure 5 illustrates the use of the special boundary to represent an hexagonal cell. This boundary condition is implemented in the following way. We assume there are a total of JT bands. The incoming flux on the J'th band in the direction $(-\mu,-\eta)$ is set equal to the outgoing flux on band JT + 1 - J in the direction (μ,η). Furthermore, the spatial variation of the flux within the band is reversed.

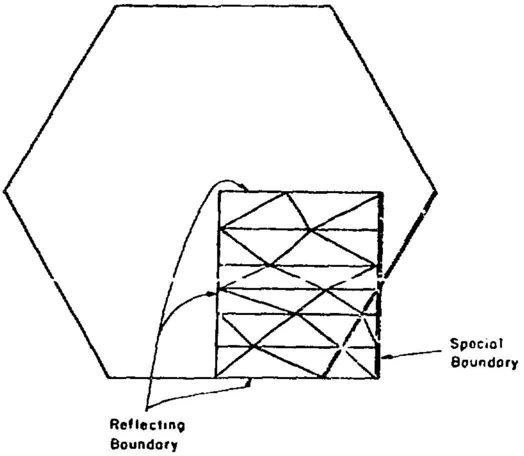


Fig. 5. Use of the special boundary condition to represent a hexagonal cell.

Hence the special boundary condition is really a 180° reflection combined with a spatial transposition about the horizontal centerline of the system.

d. Source Boundary Condition

The user of TRIPLET may specify the incoming angular flux in each quadrature direction for each of the four sides in any combination. It is not necessary when using this boundary condition that sides be parallel to coordinate axes. Because it is therefore impossible to guarantee which directions are incoming along a certain boundary, and because incoming direction at one point on the boundary may be an outgoing direction at other points, the user must enter a value for all directions at every point along a source boundary. (An option is provided whereby the user need enter only one value per triangle.) TRIPLET then selects and uses only the values corresponding to incoming directions. Boundary sources represent a source of particles and are treated as sources by the code.

2. Mesh Sweeps

In an orthogonal grid code like TWOTRAN, the mesh sweeps are usually organized so that a quadrant of directions is treated simultaneously. This can be done because neutrons streaming in all directions within a single quadrant move through the spatial mesh in the same order. Consider, for example, the positive quadrant ($\mu > 0$ and $\eta > 0$). For all directions in this quadrant, the left and bottom boundaries of each rectangular mesh cell are incoming boundaries and the top and right boundaries are outgoing boundaries. Therefore the spatial mesh can be solved in a straightforward manner beginning with the lower left corner and ending with the upper right corner.

The manner in which a triangular mesh must be swept is less simple. First, for a given direction the order in which the spatial mesh is solved is no longer straightforward but involves testing the direction of flow across triangle boundaries. Second, this order in which the mesh is swept is not the same for all directions within a quadrant but may be different for each direction. These tests must therefore be repeated for every quadrature direction (and for every group, of course).

The regular triangular mesh used by TRIPLET simplifies this testing. A regular triangular mesh retains a regular mesh structure in the y dimension so that the order in which the bands are solved is straightforward. If $\eta > 0$, the neutron flow is upward and the bands are solved in order beginning with the bottom band. If $\eta < 0$, this order is reversed. If $\eta = 0$, there is no vertical flow and thus no coupling between bands; the bands may in this case be solved in any order. For these reasons we need only consider the order in which the triangles are solved on a single band.

Let us assume that μ is positive. The direction of flow across the left boundary of the left-most triangle is then computed. The direction of flow across this boundary does not influence the order in which the band is solved but is needed to determine whether the left-most triangle is of orientation 1 or of orientation 2. The direction of flow across the right side of each triangle is then computed, moving from left to right until an outgoing right boundary is encountered. The first triangle encountered whose right boundary is outgoing can then be solved, because the flux on all incoming boundaries is known. After the fluxes in this triangle have been determined, all triangles to the left can be solved in order moving from right to left. This process of skipping several triangles and then solving from right to left is continued until the flux on all triangles on the band has been determined. If $\mu < 0$, just the reverse of the above procedure is followed. This procedure is illustrated in Fig. 6 for a typical band of triangles. The numbers indicate the order in which the triangles are solved, and the arrows indicate the direction of particle flow.

The direction of flow across the side of a triangle is not very difficult to compute. Let us assume that the x coordinate of the upper node is x_u and the x coordinate of the lower node is x_ℓ (the

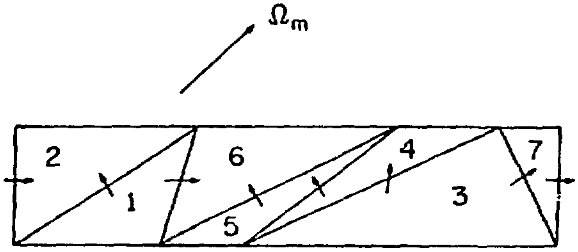


Fig. 6. Typical order of solution on a band of triangles for direction $\underline{\Omega}_m$.

triangle side is a straight line connecting these two nodes). We assume the band width is Δy. The following quantity is then computed:

$$A = \mu\Delta y + \eta(x_{\ell} - x_u).$$

If $A > 0$, the flow across this boundary is from left to right. By left to right we mean that neutrons stream from the triangle to the left of the boundary into the triangle to the right of the boundary. This may occur when $\mu < 0$ and the flow is in a leftward direction. If $A < 0$, the flow across this boundary is from right to left.

In TRIPLET the space-angle mesh is swept in the following way. The first quadrature direction is selected and the entire space mesh is solved for this direction in the manner specified above. The same process is then repeated for each of the remaining quadrature directions in turn. Because TRIPLET does not treat curved geometries, there is no angular derivative in the transport equation and the quadrature directions can be solved in any order. Note however that the presence of reflective boundaries does place certain requirements on the order in which a quadrature must be input to the code. These requirements are discussed in Sec. III. B.4.

3. Adjoint Problems

The TRIPLET program solves the adjoint transport equation by transposing the scattering and fission matrices and inverting the group order of the problem. The solution of the resulting problem in the direction $\underline{\Omega}$ is then identified with the adjoint solution in the direction $-\underline{\Omega}$. Transposition of the scattering matrix converts the normal, predominantly downscattering problem to an upscattering problem. Group order inversion restores this downscattering dominance and eliminates unnecessary upscattering iteration.

4. Iterative Processes

We assumed in deriving the methods of Sec. II.C for solving Eq. (15) on a triangular mesh that the source S_m appearing in that equation was a known function. It is clear from Eq. (16) that, if scattering or fission is present, this source function is not known but depends upon moments of the angular flux. In TRIPLET, this source is generated in an iterative context using the latest approximations for the angular flux. For the initial iterate, a flux guess must be input that permits the generation of the source function.

Sources in TRIPLET are represented, like the angular flux, by piecewise polynomial functions discontinuous across triangle boundaries. A Lagrange representation of these sources is used, so that a source is computed at each of the node points within a triangle (see Fig. 2). The spatial variation of the source within the triangle is given by the low-order polynomial interpolating these values. An exception to this strategy occurs when the continuous linear method is used to save core storage. Here a single cell-average source is computed for each triangle, and the spatial variation of the source cannot be determined.

In the following analysis we develop the iterative strategies used in TRIPLET for solving the discrete transport equation by writing these strategies for the analytic multigroup equations. We believe that details of the iteration process are clearer when presented in this manner, but the reader must keep in mind that all implied operations are actually performed in the discrete domain by methods described earlier in this report.

The multigroup transport equations can be written in operator notation as

$$L\vec{\Psi} + \Sigma\vec{\Psi} = (S_s + S_d + S_u)\vec{\Psi} + F\vec{\Psi} + \vec{Q} \quad , \qquad (28)$$

where the matrix operators L, Σ, S_s, S_d, S_u, and F represent streaming, absorption, in-group or self-scattering, downscattering, upscattering, and fission processes, respectively. The g'th elements of the vectors $\vec{\Psi}$ and $\vec{Q}$ contain the unknown fluxes and the sources, respectively, in the g'th energy group. The form of the operators appearing in Eq. (28) can be inferred by comparing that equation with Eq. (9).

TRIPLET uses a dual iteration strategy for solving the discrete analog of Eq. (28). The two nested iterations are referred to as outer and inner iterations. The outer iteration represents a sweep through all the groups, while the inner iteration is performed within each energy group. Let us assume that the angular flux $\vec{\Psi}^k$ is available from a previous outer iteration or from the input flux guess, if $k = 0$. The outer iteration then takes the form

$$L\vec{\Psi}^{k+1} + \Sigma\vec{\Psi}^{k+1} = (S_s + S_d)\vec{\Psi}^{k+1} + (S_u + F)\vec{\Psi}^k + \vec{Q} \quad . \tag{29}$$

Note that upscatter and fission sources are computed from the old flux $\vec{\Psi}^k$ but that self-scatter and downscatter sources are computed using the new flux $\vec{\Psi}^{k+1}$.

We can solve Eq. (29) for this new flux in the following manner. We first note that the matrix S_d is lower triangular, so that if the groups are solved in order beginning with the first group this term causes us no difficulty. That is, the downscatter source into group g involves only the new flux in groups g' such that g' < g. An effective source Q_g^k to the g'th group can then be computed as

$$Q_g^k = (S_d\vec{\Psi}^{k+1})_g + (S_u\vec{\Psi}^k)_g + (F\vec{\Psi}^k)_g + (\vec{Q})_g \quad , \tag{30}$$

where the notation $(\ )_g$ signifies the g'th component of the vector in parentheses. Having calculated Q_g^k, we must solve the following equation for the new flux Ψ_g^{k+1} in the g'th group

$$L_g\Psi_g^{k+1} + \Sigma_g\Psi_g^{k+1} - S_{sg}\Psi_g^{k+1} = Q_g^k \quad . \tag{31}$$

The operators L_g, Σ_g, and S_{sg} represent the g'th component of the diagonal matrix operators L, Σ, and S_s. The above equation cannot be solved easily because of the presence of the self-scatter term, which couples all directions. The methods developed in Sec. II.B and Sec. II.C are capable of solving the discrete form of Eq. (31) if scattering sources are assumed known. Thus a second iteration, the inner iteration, is suggested. In TRIPLET, this iteration takes the form

$$(L_g + \Sigma_g)\Psi_g^{k+1,\ell+1} = S_{sg}\Psi_g^{k+1,\ell} + Q_g^k \quad , \tag{32}$$

where the index ℓ is the inner iteration counter. The discrete form of the operator $L_g + \Sigma_g$ can be inverted easily by a sweep through the space-angle mesh as described earlier in this section.

The inner and outer iterations have been described above for an inhomogeneous source problem. The inner iteration remains unchanged for a k_{eff} problem, but the outer iteration is altered slightly. In place of Eq. (29) we have

$$L\vec{\Psi}^{k+1} + \Sigma\vec{\Psi}^{k+1} = (S_s + S_d)\vec{\Psi}^{k+1} + (S_u + \frac{F}{\kappa_k})\vec{\Psi}^k \quad . \tag{33}$$

In the above equation we have divided the fission source by the parameters κ_k. These parameters are computed as

$$\kappa_k = \frac{\langle F\vec{\Psi}^k\rangle}{\langle F\vec{\Psi}^{k-1}\rangle}\kappa_{k-1} \quad ,$$

with $\kappa_o = 1$ and <·> representing an integration over group, angle and space variables. The parameters κ_k approach k_{eff} for the system:

$$\kappa_k \rightarrow k_{eff} \text{ as } k \rightarrow \infty \quad .$$

5. Convergence Acceleration Methods

In most problems the inner and outer iterations described in the previous section of this report converge rapidly. There exist problems, however, for which these algorithms require excessive iteration for convergence to a satisfactory degree of accuracy. The TRIPLET program uses the acceleration method known as rebalance to accelerate the convergence of both inner and outer iterations. The code user may select one of two versions of rebalance. We call these two versions of rebalance fine mesh and whole system rebalance. The application of these convergence acceleration methods to the inner and outer iterations is discussed below.

a. Inner Iteration Rebalance

When Eq. (15) is multiplied by w_m, summed over all M directions, and integrated over the i'th triangle we obtain the following equation:

$$-FI_i^{(1)} - FI_i^{(2)} - FI_i^{(3)} + FO_i + AB_i = Q_i \quad . \tag{34}$$

The quantities appearing in the above equation are defined as follows:

$FI_i^{(k)}$ The inward partial flow of neutrons across the k'th boundary of the i'th triangle.

FO_i The total outward partial flow of neutrons across all three boundaries of the i'th triangle.

AB_i The total absorption rate of neutrons in the i'th triangle.

Q_i The total source of neutrons in the i'th triangle to the particular group at hand from all other groups and from inhomogeneous sources. Self-scatter sources are not included.

We note that self-scatter sources do not enter Eq. (34), which is a balance equation for the i'th triangle. We can obtain a balance equation for each triangle. These balance equations are satisfied if the flows and absorptions appearing in them are computed with a fully converged flux. The balance equations will not usually be satisfied by the unconverged fluxes obtained at some stage of the inner iteration.

The object of the fine mesh rebalance acceleration method is to find a set of rebalance factors f_i, one for each triangle, by which all fluxes are multiplied so that the balance equations are satisfied. We assume, in particular, that the absorption and all outflows are multiplied by the rebalance factor in the given triangle. Inflows, which are really outflows from other adjacent triangles, are multiplied by the rebalance factors corresponding to the triangles from which the flow originates. We then obtain a set of equations for the rebalance factors f_i which take the form

$$-FI_i^{(1)} f_{i_1} - FI_i^{(2)} f_{i_2} - FI_i^{(3)} f_{i_3}$$

$$+ (FO_i + AB_i) f_i = Q_i \quad . \qquad (35)$$

The subscripts i_1, i_2, and i_3 designate the three triangles that are adjacent to the i'th triangle. If the i'th triangle is adjacent to a system boundary, then the above equation must be altered in a manner which depends upon the boundary condition.

With a consistent ordering of the triangles, Eq. (35) represents a block tri-diagonal system of equations for the rebalance factors f_i. This system of equations is solved iteratively in TRIPLET, and the resulting rebalance factors are applied to all the fluxes.

Whole system rebalance is similar to fine mesh rebalance, except that a single rebalance factor is determined by which all fluxes are multiplied. This factor is easily seen to be the ratio of the total source to the sum of the leakage plus the absorption.

We observe experimentally that fine mesh rebalance almost always yields a large reduction in computing time over that required for the unaccelerated iteration. Fine mesh rebalance is usually more efficient than whole system rebalance, although it requires more core storage for the flows and absorptions that appear in Eq. (35). There are isolated problems in which the use of fine mesh rebalance can lead to an unstable algorithm. This instability is often connected with use of the continuous linear method. If the user encounters such difficulty, he is advised to use whole system rather than fine mesh rebalance.

Two methods have been incorporated in TRIPLET for the stabilization of fine mesh rebalance. The first method checks the set of Eqs. (35) for diagonal dominance. Diagonal dominance of this set of equations is guaranteed by adding factors to the diagonal entries and to the sources. This procedure changes the rebalance factors so that they are closer to unity. The second method adds another factor to both the diagonal and the source. This factor is chosen to be a small fraction of the largest diagonal element. This second method also forces the rebalance factors to be closer to unity and tends to damp oscillations that sometimes develop in the application of this method.

b. Outer Iteration Rebalance

To accelerate the convergence of the outer iteration process, we determine a different set of scale factors, f. We collapse the entire group structure to a single group, accumulating the flows from each group. We also calculate the total absorption using an effective σ_{ag} calculated from the input cross sections by

$$(\sigma_{ag})_{eff} = \sigma_{tg} - \sum_{h=1}^{IGM} \sigma_{sog\to h} \quad .$$

When we are performing an adjoint calculation we must, because of the cross-section transposition, calculate a different effective absorption,

$$(\sigma_{ag})_{eff} = \sigma_{tg} - \sum_{h=1}^{IGM} \sigma_{soh\to g} \quad .$$

The source for over-all-group rebalance consists of the group sum of the inhomogeneous source (if any) and the group sum of the fission source (if any). If there is a fission source, we perform a source iteration to determine the rebalance factors. If there is no inhomogeneous source, this iteration can also be used to estimate an eigenvalue, say k_{eff}. We choose to do this only for the first two outer iterations, thereafter considering the fission source to be a known inhomogeneous source.

The outer iteration rebalance process is advantageous because it accelerates all types of problems, e.g., inhomogeneous source problems with upscatter and/or fission, or eigenvalue problems with or without upscatter.

In both the within-group and over-all-group rebalancing calculations, the scale factors all usually approach unity in a few inner or outer iterations.

6. Convergence Tests

There are three levels of iterative processes in the TRIPLET program: (1) the inner iteration, in which the within-group scattering source or the boundary flux at an implicit boundary changes, (2) the outer iteration in which the fission or upscattering source changes or which is caused by artificial inner iteration limitation (usually in inhomogeneous source problems), and (3) the parametric eigenvalue search iteration in which, after a converged outer iteration, the value of a material concentration or a time absorption is changed. Two additional iterations are required for the calculation of fine-mesh rebalance factors, one for the factors themselves and one for the fine-mesh rebalance eigenvalue.

Two convergence precisions are input: EPS and XLAX. In a parametric eigenvalue search, two values of (r indicates iteration number)

$$\lambda^{(r)} = \frac{\text{Fission Source}^{r} + \text{Inhomogeneous Source}}{\text{Fission Source}^{r-1} + \text{Inhomogeneous Source}} \qquad (36)$$

are required to differ by less than XLAX before a new eigenvalue guess is computed. All other processes are tested against precisions derived from EPS. These precisions are

EPSO = EPS		outer iteration convergence
EPSI = EPS		inner iteration convergence
EPSR = 10^{-4}		fine-mesh rebalance numbers
EPSX = EPS		collapsed group fine-mesh convergence, both factors (10*EPSX) and eigenvalue (EPSX).

In the inner iteration process, we require

$$\max_i \left| 1 - \frac{\phi_i^{r-1}}{\phi_i^{r}} \right| < \text{EPSI}$$

for the r'th iteration where ϕ_i is the average scalar flux in the i'th triangle before application of the rebalance factors. If the number of iterations is greater than the input value IITL, the inner iteration is terminated.

For the iterative computation of the fine mesh rebalance factors in the inner iteration, we require

$$\max_i \left| 1 - \frac{f_i^{r-1}}{f_i^{r}} \right| < \text{EPSR} \quad .$$

In the outer process, if both

$$|\lambda - 1| < \text{EPSO and } \max_i |1 - f_i| < 10*\text{EPSX} \quad ,$$

the problem is terminated after one final outer iteration. In the fine-mesh iterative process (denoted with iterative superscript r) we require

$$\left|\lambda^{(r)}_{\text{Fine Mesh}} - 1 \right| < \text{EPSX} \quad ,$$

if IEVT = 1 (k_{eff} calculations) and

$$\left|1 - \lambda^{(r)}_{FM} / \lambda^{(r-1)}_{FM}\right| < \text{EPSX} \quad ,$$

if IEVT ≠ 1 (parametric eigenvalue or inhomogeneous source plus fission problems).

III. A GUIDE TO USER APPLICATION

In this section we provide information needed by the user to understand TRIPLET options and to prepare input for the code.

A. Overall Program Flow

A schematic flow chart for TRIPLET is given in Fig. 7. A more detailed flow chart is provided below in the section on programming information.

B. Details of Program Options

1. Cross Sections

a. Input Formats

The TRIPLET program accepts cross sections either from the standard file ISOTXS,[4] in FIDO format,[5] or in the standard Los Alamos format. In upscattering problems, the program does not need the special σ^{up} cross section which is required in earlier Los Alamos programs.[6] In TRIPLET, it is assumed that σ^{up} is present, and σ^{up} is automatically removed from card input cross-section sets unless the user tags the input number IHT with a minus sign. Cross sections read with the FIDO format may not contain σ^{up}.

The Los Alamos cross-section format assumes that each nuclide is described by a block of cross sections of IHM rows for IGM group columns. The row position of cross sections is specified relative to the total cross section, σ_t (row IHT), and the within-group scattering cross section, $\sigma_{s,g\to g}$, (row IHS). It is assumed that the row order of the cross sections is as follows:

	Row	Cross-Section Type-Group g
	.	
	.	
	.	
	IHT-4	$\sigma_{n,2n}$
	IHT-3	σ_{tr}
	IHT-2	σ_a
	IHT-1	$\nu\sigma_f$
	IHT	σ_t
	IHT+1	$\sigma_{s,g+N\to g}$
	.	
IHM	.	
	.	
	IHS-2	$\sigma_{s,g+2\to g}$
	IHS-1	$\sigma_{s,g+1\to g}$
	IHS	$\sigma_{s,g\to g}$
	IHS+1	$\sigma_{s,g-1\to g}$
	IHS+2	$\sigma_{s,g-2\to g}$
	.	
	.	
	.	
	IHS+M	$\sigma_{s,g-M\to g}$

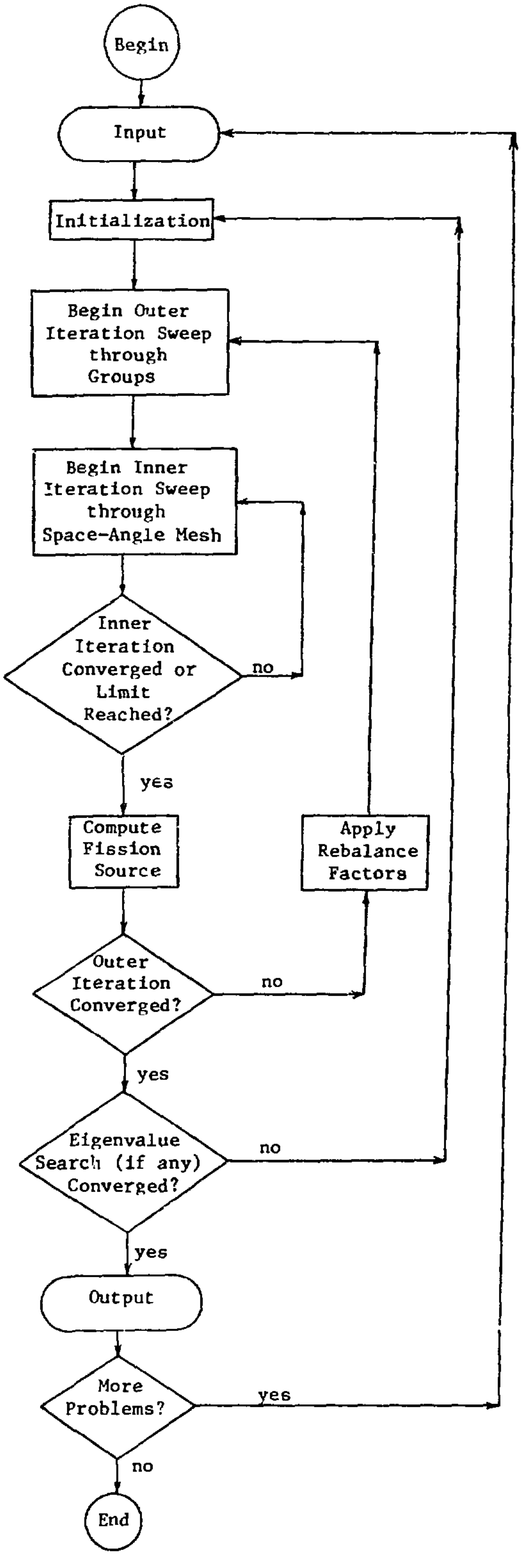

Begin
Input
Initialization
Begin Outer Iteration Sweep through Groups
Begin Inner Iteration Sweep through Space-Angle Mesh
Inner Iteration Converged or Limit Reached?
no
yes
Compute Fission Source
Apply Rebalance Factors
Outer Iteration Converged?
no
yes
Eigenvalue Search (if any) Converged?
no
yes
Output
More Problems?
yes
no
End

Fig. 7. Simplified logical flow diagram for TRIPLET.

In this format, group g+1 corresponds to a group of lower energy than group g. The symbol $\sigma_{s,g-2\rightarrow g}$ denotes the scattering transfer probability from group g-2 to group g. The format allows N groups of upscatter and M groups of downscatter; i.e., the scattering matrix need not be symmetric. However, all cross-section blocks must have the same values for IHM, IHS, and IHT. The fission cross section, σ_f, times the mean number of neutrons per fission, ν, must be located in row IHT-1, and the absorption cross section, σ_a, must be entered in row IHT-2. If a scattering matrix is used to represent (n,2n) reactions, that is, if $\sigma_{s,g\rightarrow h}$ contains 2 times the (n,2n) transfer probability $\sigma_{n,2n;g\rightarrow h}$, then the value,

$$\sigma_{n,2n} = \sum_{\text{all } h} \sigma_{n,2n;g\rightarrow h} \quad ,$$

must be entered in IHT-4. The user is free to enter additional cross sections at the top of the format. These extra cross sections are not used in the calculation, but are available for reaction-rate computations after the particle flux is obtained. It is sometimes convenient to locate the transport cross section, σ_{tr}, in IHT-3 and use this cross section instead of σ_t in the calculation of buckling corrections.

b. Cross-Section Mixing

The user is free, in TRIPLET, to enter macroscopic cross sections and bypass the mixing algorithms; specification of the input value MS = 0 is all that is required for this. If MS ≠ 0, the user must provide three sets of MS numbers which are stored in the vectors MIXNUM, MIXCOM, and MIXDEN. These numbers are used in the following algorithm to manipulate cross sections blocks:

```
    DO  315  M = 1, MS
    N = MIXNUM(M)
    L = MIXCOM(M)
    AD = MIXDEN(M)
    DO  315  I = 1, IHM
    IF(L.EQ.0) GO TO 310
    IF((AD.EQ.0.0).AND.(IEVT.EQ.3)) GO TO 313
    C(I,N) = C(I,N) + AD*C(I,L)
    GO TO 315
313 C(I,N) = EV*C(I,N)
    GO TO 315
310 C(I,N) = AD*C(I,N)
315 CONTINUE
```

In this algorithm, cross-section block N is created or altered by adding multiples of block L or by multiplying the block N by a factor. Let us consider some examples.

Suppose we have entered 45 cross sections as input. Then any mixtures that are made must be given block numbers higher[*] than 45. Suppose we enter:

MIXNUM (N)	MIXCOM (L)	MIXDEN (AD)
46	0	0.0
46	1	0.0478
46	20	0.0333
47	0	0.0
47	2	0.75
47	3	0.25
47	0	0.1179
48	0	0.0
48	15	0.0049
48	14	0.0078
48	48	0.0
49	0	0.0
49	33	0.5
49	34	0.5
49	0	0.187
49	49	0.0
49	46	0.1

For this example we have MS = 17 instructions. In the first three instructions, block 46 is cleared (set to zero) and then made up of 0.0478 parts of block 1 and 0.0333 parts of block 20. If block 1 and 20 are microscopic cross sections in barns, then 0.0333 and 0.0478 times 10^{24} are the atomic densities. In the second set of instructions, block 47 is cleared and then made up of 0.1179 times the result of adding

[*] To preserve the input values. If these need not be saved, mixtures can be created in lower block numbers.

three-fourths of block 2 to one-fourth of block 3. In the next set of instructions, block 48 is cleared and made up of portions of blocks 15 and 14. If IEVT (the input eigenvalue type option) is 3, then the resulting block 48 is multiplied by EV (the input eigenvalue guess). In this type of problem the program attempts to find a value of EV such that the resulting concentration of block 48 renders the system critical. If IEVT ≠ 3, the line of instructions 48, 48, 0.0 would not alter the composition of block 48. In the final sequence, block 49 is made up of 0.187 times one-half of block 33 and block 34, and provision is made to search for the concentration of this portion of 49 to which is always added 0.1 of the previously mixed block 46. It should be clear that there are many possibilities not covered in these examples, but by examining the FORTRAN instructions above, the user should be able to prepare his own sets of mixture instructions.

c. Anisotropic Cross Sections

In the TRIPLET program it is assumed that the scattering transfer probability can be represented by a finite Legendre polynomial expansion; i.e., that

$$\sigma_s(E' \to E,\mu_o) = \sum_{n=0}^{ISCT} \frac{2n+1}{4\pi} P_n(\mu_o)\sigma_{sn}(E' \to E) \quad , \tag{37}$$

where ISCT is an input control integer. Thus if ISCT > 0, additional blocks of scattering transfer cross sections must be entered for those nuclides for which anisotropic scattering sources are to be computed. In these blocks, the rows 1 through IHT are zero, and $\sigma_{sn,g\to h}$ (the energy average of $\sigma_{sn}(E' \to E)$ in groups g and h) is entered as for the isotropic component of the cross section. It is assumed in TRIPLET that blocks of anisotropic cross sections which are used in the calculation have block numbers in ascending sequence, starting with the isotropic cross-section block. For example, suppose that block 50 is the isotropic cross-section block for hydrogen and that ISCT = 3. Then, block 51 must be σ_{s1} for hydrogen, block 52 must be σ_{s2} and block 53 must be σ_{s3}. If a material is made by mixing two anisotropic scatterers, then the anisotropic blocks must also be mixed with the same densities to form anisotropic blocks for the material.

Anisotropic scattering sources may be computed selectively within each zone, but in all zones in which such sources occur, the number of anisotropic scattering blocks (ISCT) must be the same.

d. Adjoint Cross Sections

In adjoint calculations, cross sections are entered just as for a direct calculation. The program then transposes the scattering matrices and, because this usually changes a downscattering problem to an upscattering problem, reverses the group order of the blocks. Further, the effective absorption in an adjoint calculation is not simply related to σ_a. That is, the effective absorption is normally

$$(\sigma_a)_{eff} = \sigma_t - \sum_{\text{all } h} \sigma_{so,g\to h} \quad . \tag{38}$$

But when the scattering matrix has been transposed, the effective absorption is

$$(\sigma_a)_{eff} = \sigma_t - \sum_{\text{all } h} \sigma_{so,h\to g} \quad .$$

In regular and adjoint problems, these quantities are calculated and processed in mixing operations for use in the rebalancing algorithms.

e. Cross-Section Checking

As input cross sections are processed, Eq. (38) is computed and compared to the input value of σ_a. If the relative difference between the input total cross section and the computed total cross section exceeds EPS (outer convergence precision), the user is so informed.

2. Finite Element Approximation Specification

The finite element spatial discretization is discussed in detail in Sec. II.C. The nature of the spatial approximation is determined by two input parameters, NP and ISDS. These two parameters provide the TRIPLET user considerable freedom in the selection of the spatial approximation.

The quantity NP is the order of the polynomial representation of the angular flux and may be chosen to be any positive integer (NP ≥ 1). For an NP'th order polynomial, NPT = (NP + 1)(NP + 2)/2 angular fluxes must be determined on each triangle on each sweep through the space-angle mesh. Because NPT increases rapidly with NP, the higher order methods are expensive in terms of computing time. Furthermore, the solution algorithm for the linear case (NP = 1) has been extensively optimized. The linear option is recommended for most problems in which computation time is a significant consideration.

The parameter ISDS selects either the continuous or discontinuous form of the spatial approximation. For ISDS = 0, continuity of the angular flux across triangle boundaries is enforced in the manner discussed in Sec. II.C. For ISDS = 1, discontinuity of the angular flux across triangle boundaries is allowed. The ISDS = 0 option (continuous method) is provided to minimize core storage requirements for large problems, with only a slight reduction in accuracy. The discontinuous method requires the storage of the scalar flux and moments at all NPT points on each triangle whereas the continuous method requires storage of only the cell-average scalar flux and moments on each triangle. Furthermore, the solution algorithm for the continuous method is simpler than for the discontinuous method, resulting in a 10 - 20% reduction in computation time. The continuous method is available only for the linear (NP = 1) order approximation.

3. Geometry and Boundary Condition Satisfaction

a. Spatial Mesh

The input of information necessary to specify the spatial mesh is greatly simplified by the restriction to the regular triangular mesh discussed in Sec. I. The first three items necessary for complete specification of the mesh are:

(1) The number of bands of triangles: JT.

(2) The number of triangles on each band: IT array, JT entries, from the bottom band to the top band.

(3) The orientation of the first triangle on each band: ITT array, JT entries, from the bottom band to the top band.

With the above data, the subroutine KSET calculates the total number of triangles (NTC), the total number of triangle vertices (NTX), the number of upward pointing triangles on the bottom band (NUPB), the number of downward pointing triangles on the top band (NDNT), and two required indexing arrays

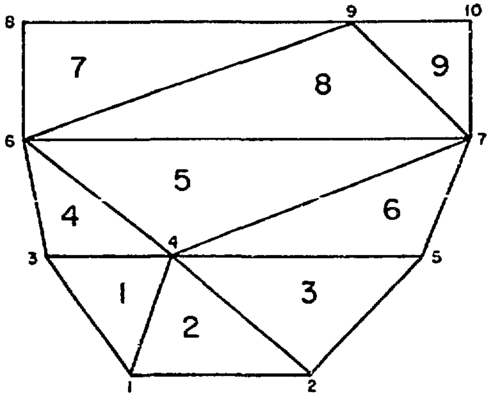


Fig. 8. Example of x-coordinate (small numbers) and triangle (large numbers) indexing.

(KC and KX). The remaining mesh specification items which must be input are:

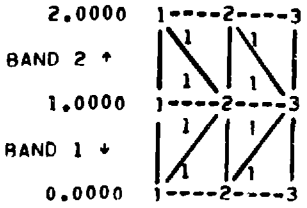
(4) The height of each triangle band: HY array, JT entries, from the bottom band to the top band.

(5) The x-coordinates of the triangle vertices: x array, NTX entries.

(6) The cross-section material identification number for each triangle: IDCS array, NTC entries.

The x-coordinates are numbered sequentially from the lower left corner to the upper right corner, as indicated in Fig. 8, and are input in this order. Each triangle has an identification number associated with it. These triangle i.d.'s run from left to right, starting with the bottom band and proceeding upward, as illustrated in Fig. 8.

After all of the problem input has been read by TRIPLET, the above information is converted into a pictorial representation of the mesh by subroutine MAPPER. An 8 triangle example is shown.



The material identification number is printed within each triangle. For this example, the system is homogeneous in material 1. The y-coordinate is printed to the left of each node line (line of triangle vertices). The arrow beside the band index denotes the orientation of the first triangle on each band. The triangle vertices on each node line are numbered from left to right. The TRIPLET user may draw lines between the vertices to complete the picture of the spatial mesh, as illustrated in the example. Furthermore, a table of the x-coordinates is printed immediately before the material map and numbered to give the x-coordinate of the vertices along each node line. The material map is drawn to scale in the x-direction, with a uniform band height. For large meshes with many triangles, the material map is automatically continued onto additional pages. Additional characters are used to indicate those boundaries that are reflecting or special.

For input of the distributed source or flux guess with the discontinuous model the TRIPLET user can specify a pointwise variation over each triangle. The placement of each interpolation point on a triangle is shown in Fig. 2 for several low-order polynomials. The indexing of these points on a triangle depends upon whether the triangle is upward or downward pointing. For upward pointing triangles, the numbers start with the left-most point on the bottom row and proceed to the right, from the bottom row of points to the last point on the top vertex. For downward pointing triangles, the numbers start with the right-most point on the top row and proceed to the left, from the top row of points to the last point on the bottom vertex. This is illustrated in Fig. 9 for a second order (NP = 2) polynomial example.

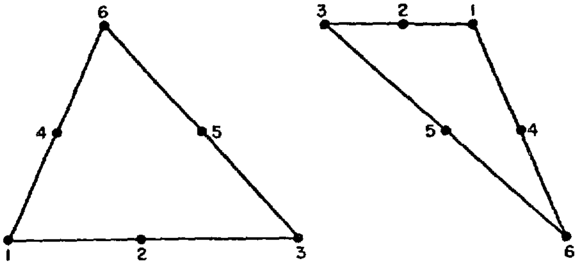


Fig. 9. Indexing of points on a triangle for NP = 2.

b. Boundary Conditions

The TRIPLET user must select one of the following four boundary conditions for each of the system boundaries.

(1) Vacuum boundary conditions - the angular flux on the boundary is set to zero for all incoming directions.

(2) Reflective boundary conditions - the angular flux on the boundary for incoming directions is set equal to the outgoing flux in the direction corresponding to specular reflection.

(3) Boundary sources - the angular flux on the boundary for incoming directions is set equal to a user-specified value.

For the right boundary, the user is allowed the

(4) Special boundary condition - the angular flux on the boundary of the j'th band for incoming directions $(-\mu,\eta)$ is set equal to the outgoing flux on band JT-j+1 for directions $(\mu,-\eta)$.

Use of reflecting or special boundary conditions imposes certain requirements on the S_n quadrature set, as discussed in Sec. III.B.4. Side boundaries must be vertical if reflecting or special boundary conditions are applied. For a special boundary condition, the band heights must be symmetric about the centerline ($HY_{JT-j+1} = HY_j$).

4. Angular Quadrature Coefficient Specifications.

The TRIPLET user has the option of obtaining the angular quadrature coefficients from interface file ISNCON or from a built-in set in subroutine SNCON, or the user may enter these coefficients as card input. The input parameter ISN is used to specify the source of these coefficients.

The built-in constants are the standard S_n constants of TWOTRAN-II.[1] If one of these built-in sets is used, the total number of quadrature angles, MM, must be entered as 4, 12, 24, 40, 60, 84, 112, or 144, corresponding to S_2, S_4, ..., S_{16}. The ordering of the built-in sets is illustrated in Fig. 10 for the S_6 constants.

Considerable freedom is allowed the TRIPLET user in the input of quadrature sets. Two requirements must be met. The weights, w_m, must be

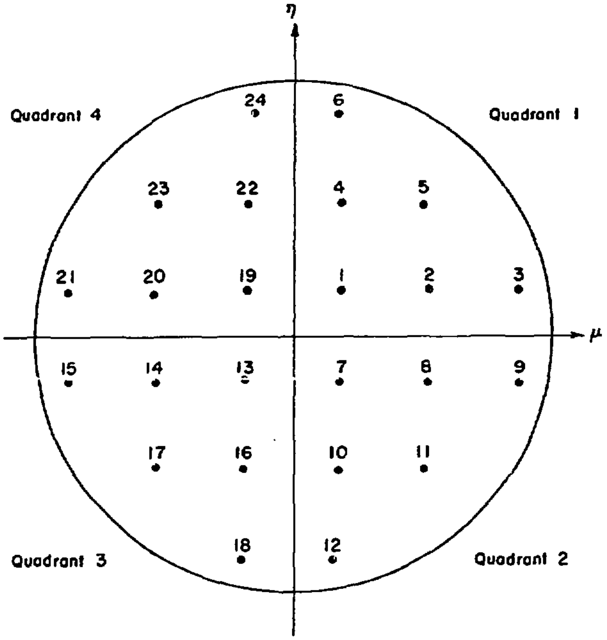


Fig. 10. S_6 quadrature directions.

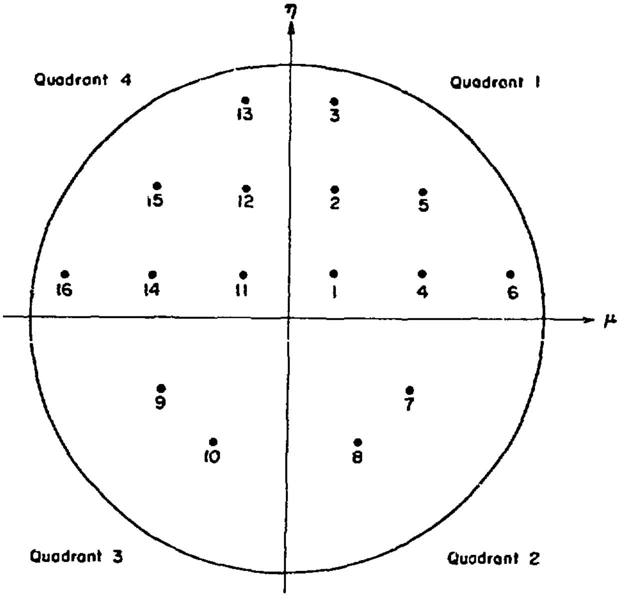


Fig. 11. Example of quadrature set for left or right reflecting boundary.

correctly normalized $\left(\sum_{m=1}^{MM} w_m = 1\right)$, and the quadrature set must correctly integrate the first ISCT moments. For vacuum and source boundary conditions, the ordering and symmetry of the directions are not required to conform to that of the standard built-in set. For reflecting and special boundary conditions, the following symmetry conditions must be satisfied:

(a) Top or bottom reflecting boundary condition - the directions must possess symmetry about the μ axis.

(b) Left or right reflecting boundary condition - the directions must possess symmetry about the η axis.

(c) Special boundary condition - the directions must possess symmetry about both the μ and η axes. For hexagonal cell calculations using the special boundary condition, the left, top, and bottom sides must be reflecting.

In addition to symmetry requirements for reflecting and special boundary conditions, the quadrants must be entered in the order shown in Fig. 10. That is, all ordinates for $\mu > 0$, $\eta > 0$ are entered first; then ordinates for $\mu > 0$, $\eta < 0$; then ordinates for $\mu < 0$, $\eta < 0$; and ordinates for $\mu < 0$, $\eta > 0$ last. In addition, the directions within a quadrant must be "consistent," as described below. An example of a special quadrature set for use with either left or right reflecting boundaries is illustrated in Fig. 11.

"Consistent" ordering of the quadrature directions is defined as that ordering which permits the correct operation of the indexing algorithm for storage of the outgoing angular flux on a reflecting or special boundary. The indexing algorithm is based on the four quantities NPQ1, NPQ2, NPQ3, and NPQ4; the number of points (or directions) in quadrants 1, 2, 3, and 4, respectively. These quantities are calculated in subroutine SNCON. The ordering is "consistent" for

(a) reflecting top or bottom boundaries if

$$\begin{aligned} \mu_{m+NPQ1} &= \mu_m , \\ \eta_{m+NPQ1} &= -\eta_m , \end{aligned} \qquad \text{for m in quadrant 1,}$$

$$\begin{aligned} \mu_{m-NPQ4} &= \mu_m , \\ \eta_{m-NPQ4} &= -\eta_m , \end{aligned} \qquad \text{for m in quadrant 4,}$$

(b) reflecting left or right boundaries if

$$\begin{aligned}\mu_{m+NPQ1+NPQ2+NPQ3} &= -\mu_m ,\\ \eta_{m+NPQ1+NPQ2+NPQ3} &= \eta_m ,\end{aligned} \quad \text{for m in quadrant 1,}$$

$$\begin{aligned}\mu_{m+NPQ2} &= -\mu_m ,\\ \eta_{m+NPQ2} &= \eta_m ,\end{aligned} \quad \text{for m in quadrant 2 .}$$

For example, consider the quadrature set shown in Fig. 11. Here NPQ1 = 6, NPQ2 = 2, NPQ3 = 2, and NPQ4 = 6. For an outward direction, m, on the right boundary in quadrant 2, the reflected direction is m' = m + NPQ2. Thus an outward directed flux on the right boundary in the direction (μ_8,η_8) must be stored in the position for the inward direction (μ_{10},η_{10}).

5. Source Options

The TRIPLET user may specify an anisotropic distributed source or the boundary flux at any boundary of the system. The inhomogeneous distributed source must be represented by the finite spherical harmonic expansion

$$Q_g(\underline{r},\mu,\eta) = \sum_{n=0}^{IQAN} 2n+1 \sum_{k=0}^{n} R_n^k(\mu,\phi(\mu,\eta))Q_{gn}^k(\underline{r}) \ , \qquad (39)$$

in which IQAN is an input number designating the order of anisotropy of the source, and R_n^k is the spherical harmonic defined in Eq. (4) above. In these terms,

$$Q_{gn}^k = \frac{1}{4\pi} \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\phi Q_g(\underline{r},\mu,\eta)R_n^k(\mu,\phi(\mu,\eta)) \quad .$$

We have written the integral in this equation over the entire range of ϕ and used a 4π normalization of the integral, but it should be clear that Q_g must be symmetric in $\phi[Q_g(\phi) = Q_g(-\phi)]$, and this we have noted by writing Q_g as a function of μ and η alone.

In addition to specifying IQAN, the user must enter [(IQAN + 1)(IQAN + 2)]/2 components of Q_g multiplied by (2n + 1); that is, the user must enter

$$(2n + 1)Q_{gn}^k \quad \begin{aligned} k &= 0, 1, \ldots, n\\ n &= 0, 1, \ldots, \text{IQAN}\end{aligned}$$

in the order shown in Table I.

TABLE I

ORDERING OF ANISOTROPIC DISTRIBUTED SOURCE COMPONENTS

Component Number	n	k
1	0	0
2	1	0
3	1	1
4	2	0
5	2	1
6	2	2
7	3	0
8	3	1
9	3	2
10	3	3
.		
.		
.		
etc.		

When using the anisotropic distributed source option, the order of anisotropic scattering, ISCT, must be at least as large as IQAN so that the requisite number of scattering coefficients R_n^k is computed.

The TRIPLET user is also allowed to specify the value of the incoming flux along any or all boundaries. If the source boundary condition integer is -3, the boundary flux is assumed to be constant along the face of each boundary triangle (subsequently referred to as the flat source mode). If the source boundary condition integer is +3, the boundary flux is assumed to have a pointwise variation along the face of each boundary triangle (subsequently referred to as the shaped source mode). Use of the source boundary condition integer as -3 reduces (by 1/(NP + 1)) the number of entries required to specify the boundary flux. The boundary sources, summed over all groups and integrated over all angles and surface area, are added to the volume-energy integrated distributed sources (if any) for the normalization of all inhomogeneous sources.

6. Source and Flux Input Options

a. Source Input Options

If a distributed source of anisotropy IQAN is designated, the NMQ = (IQAN+1)(IQAN+2)/2 components of the source must be entered for each group in the order listed in Table I. This may be accomplished with the use of four options designated by the input value of IQOPT and described below. The various source arrays described below are supplied for each group, with the distributed source (if any) specified first, followed by the boundary sources (if any). The ordering of the source input within each group g is:

(1) Distributed source (optional) - order of input depends on IQOPT as described below.

(2) Right boundary source (optional) - from the bottom triangle to the top triangle on the right side.

BR_{gkjm}; k=1,NPP; j=1,JT; m=1,MM for IBR = +3

or BR_{gjm}; j=1,JT; m=1,MM for IBR = -3.

(3) Bottom boundary source (optional) - from the left triangle on the bottom band to the right triangle.

BB_{gkjm}; k=1,NPP; j=1,NUPB; m=1,MM for IBB = +3

or BB_{gjm}; j=1,NUPB; m=1,MM for IBB = -3.

(4) Top boundary source (optional) - from the left triangle on the top band to the right triangle.

BT_{gkjm}; k=1,NPP; j=1,NDNT; m=1,MM for IBT = +3

or BT_{gjm}; j=1,NDNT; m=1,MM for IBT = -3.

(5) Left boundary source (optional) - from the bottom triangle to the top triangle on the left side.

BL_{gkjm}; k=1,NPP; j=1,JT; m=1,MM for IBL = +3

or BL_{gjm}; j=1,JT; m=1,MM for IBL = -3.

Here NPP(=NP+1) is the number of points on one side of a triangle, NUPB is the number of upward pointing triangles on the bottom band, and NDNT is the number of downward pointing triangles on the top band. The above notation is that of standard FORTRAN (the innermost set of indices cycled most rapidly). For example, for IBL = -3, the boundary flux BL is entered in a continuous stream, the first JT numbers for m = 1, the next JT numbers for m = 2, etc.

The IQOPT parameter provides the TRIPLET user with five options for entering the distributed source. These options are:

IQOPT Option

0 No input entered. The complete distributed source array, Q_{gnki}, is automatically set to zero.

1 Enter an energy spectrum (GR_{gn}, g=1,IGM) for each anisotropic component n. The source array is then formed as $Q_{gnki} = GR_{gn}$; k=1,NPPT; i=1,NTC for each n and g.

2 Enter the complete source array as (Q_{gnki}; k=1,NPPT; i=1,NTC) for n=1,NMQ and g=1,IGM. See Sec. II.B.3.a. for the ordering of the NPPT points on each triangle and the ordering of the NTC triangles on the mesh. The above entries are input as a block of NPPT*NTC numbers for each n and g.

3 Enter first a spectrum (GR_{gn}, g=1,IGM) and then a spatial shape (F_{nki}; k=1,NPPT; i=1,NTC) for n=1,NMQ. The source array is then formed as $Q_{gnki} = GR_{gn}F_{nki}$; k=1,NPPT; i=1,NTC for each n and g. See Sec. II.B.3.a. for the ordering of the NPPT points on each triangle and the ordering of the NTC triangles on the mesh. The above entries are input as blocks of IGM numbers and NPPT*NTC numbers for each n.

5 The distributed source (but not boundary sources) is read from a standard interface file FIXSRC mounted on unit IFIXSR.

The quantity NTC is the total number of triangles in the spatial mesh. NPPT is the number of points per triangle for which the source and flux must be stored. For the continuous model (ISDS = 0), NPPT = 1, and for the discontinuous model (ISDS = 1), NPPT = NPT.

b. Flux Input Options

Options for reading an input flux guess are similar to those for reading an input source. If ISCT is the order of anisotropic scattering, then there are NM = (ISCT + 1)(ISCT + 2)/2 spherical harmonic components of the angular flux, ordered as in Table I. Options for entering these components are selected by the input value of the integer ISTART. A negative value for ISTART indicates that only the angle-integrated or scalar flux is to be read. Allowed values of ISTART are:

ISTART Option

-6 A problem restart dump is read from unit NDUMP1 and edited with a minimum of computation.

-5 Same as option +5.

-3 Same as option +3, isotropic component only.

-2 Same as option +2, isotropic component only.

-1 Same as option +1, isotropic component only.

0 No flux guess entered. A fission guess of unity in every mesh cell is automatically supplied.

+1 Enter an energy spectrum (GR_{gn}, g=1,IGM) for each anisotropic component n. The flux guess is then formed as $FLUX_{gnki} = GR_{gn}$; k=1,NPPT; i=1,NTC for each n and g.

+2 Enter the entire flux array as ($FLUX_{gnki}$; k=1,NPPT; i=1,NTC) for n=1,NM and g=1,IGM. See Sec. III.B.3.a. for the ordering of the NPPT points on each triangle and the ordering of the NTC triangles on the mesh. The above entries are input as a block of NPPT*NTC numbers for each n and g. Imagine that the fluxes are read by:

```
      DO 17 G=1,IGM
      DO 17 N=1,NM
   17 READ ((FLUX(G,N,K,I), K=1,NPPT),
         I=1,NTC)
```

+3 Enter first a spectrum (GR_{gn}, g=1,IGM) and then a spatial shape (F_{nki}; k=1,NPPT; i=1,NTC) for n=1,NM. The flux array is then formed as $FLUX_{gnki} = GR_{gn}F_{nki}$; k=1,NPPT; i=1,NTC for each n and g. See Sec. III.B.3.a. for the ordering of the NPPT points on each triangle and the ordering of the NTC triangles on the mesh. The above entries are input as blocks of IGM numbers and NPPT*NTC numbers for each n. Imagine the fluxes to be read by:

```
      DO  17  N=1,NM
      READ  (GR(G), G=1,IGM)
      READ  ((F(K,I), K=1,NPPT), I=1,NTC)
      DO  17  G=1,IGM
      DO  17  K=1,NPPT
      DO  17  I=1,NTC
   17 FLUX(G,N,K,I) = GR(G)*F(K,I)
```

+5 An entire scalar flux guess is read from standard interface file RTFLUX or ATFLUX on unit ITFLUX.

+6 A problem restart dump is read from unit NDUMP1 and computation proceeds in the normal manner.

7. Flux Dumps and Restart Procedures

The three types of dumps that are taken have the same form, and each may be used to restart a problem. A periodic dump is taken every M minutes where M is a program variable which can be set to meet particular installation requirements. A final dump is always taken after the successful completion of a problem, and a time limit dump is taken after a user-specified period of time. Dumps are written alternately on units NDUMP1 and NDUMP2 depending on which is free; an output message is written to indicate which unit contains the latest dump.

When problem execution is continued using a restart dump, certain input parameters can be changed and edit specifications can be added or modified. It is even possible to use the program to edit a dump. However, if this option is selected and more information is required to perform the edit, one more outer iteration may be required to generate and store the angular fluxes to be used during the edit.

To restart a problem, a special problem input deck, consisting of three sections, is required. The first section is the same as the normal problem integer input with the value of ISTART set to ±6. During restart all other integer values are ignored. The second section of restart input makes use of the namelist feature standard to FORTRAN to permit the user to change certain input parameters (those listed below) and to enter only those he wants to change. If no changes are desired, this section is omitted.

The special character for beginning and terminating a namelist block may vary from machine to machine. On CDC computers the dollar character is used. The first column of namelist cards is ignored. Columns two through eight must contain $TRIPIN, and column nine must contain a blank for the first card. Using entries of the free-field form: name = value, the user defines his changes, separating different entries by commas and ending the last entry with a dollar sign. Continuation cards are permitted provided that the last entry of the preceding card ends with a comma. The integers which may be changed are:

1. IITL Maximum number of inner iterations.
2. ITLIM Time limit.
3. IEDOPT Edit input indicator.
4. I2 Final flux print indicator.
5. I4 Final fission print indicator.
6. IFO Interface file output indicator.

The floating point values which may be changed are:

1. EPS Convergence precision.
2. POD Parameter oscillation damper used in eigenvalue searches.
3. XLAL Search lambda lower limit.
4. XLAH Search lambda upper limit.
5. XLAX Fine-mesh search precision.

The third section of restart input is the edit section. The composition of the edit input section is determined by the value of IEDOPT after the namelist section is read. If the original problem had IEDOPT > 0 and if IEDOPT is not changed to zero on restart, then the edit input must be re-entered.

The sign of ISTART determines the iteration path following a restart. If ISTART = +6, then computation is restarted at the exact point at which the dump was taken. For a final dump restart, computation will begin with another outer iteration and will continue until convergence is detected (remember that EPS may be changed). If ISTART = -6 and all information required for an edit exists, the final output portion of the code is executed at once with no further iteration. If this information does not exist, TRIPLET will perform the minimum amount of computation necessary to generate this information. This additional computation will involve a maximum of one complete outer iteration. Following the generation of this edit information, the final output routines are entered.

8. Rebalance Acceleration of Iterations

The user is given two choices of rebalance schemes for acceleration of the inner and outer iterations. Particle balance may be enforced over the entire system (whole-system rebalance) or each triangle (fine-mesh rebalance). Normally, more rapid convergence will be achieved with fine-mesh rebalance, at the expense of 7*NTC additional words of fast core. Experience has shown that for some problems, especially when using the continuous difference scheme, the fine-mesh rebalance algorithm is unstable. For such problems one must use whole-system rebalance to obtain a solution. When the special boundary condition is used, only whole system rebalance is permitted.

9. Eigenvalue Searches

It is possible in TRIPLET to adjust nuclide concentrations or the value of the time absorption to achieve a desired value of k_{eff}. This value is taken to be unity (criticality) unless the parametric eigenvalue trigger (IPVT) is set to unity. In this case, the parametric value of k_{eff} is entered as an input number. If IPVT=2, a 1/v absorber of value PV is added to the problem in each space cell.

The modification of cross-section concentrations takes place as indicated in Sec. III.B.1.b. This type of problem is run when the eigenvalue type indicator (IEVT) is 3. If IEVT is 2 (time absorption computation), the value EV/v_g is added to the absorption and total cross sections in each group. Here v_g is the speed associated with energy group g.

Regardless of the parameter being adjusted, the search is executed by performing a sequence of k_{eff} calculations, each for a different value of the parameter being treated as the eigenvalue. Each of the successive k_{eff} calculations is accelerated by rebalance, but the search for the desired value of k_{eff} is conducted by subroutine NEWPAR. Regardless of the nature of the problem, the search is for a value of the parameter which makes the value of λ defined in Eq. (36) unity.

In the following description of NEWPAR, it is helpful to refer to Fig. 12 in which the deviation of λ from unity is plotted against outer iteration number.

For the initial system, NEWPAR continues outer iteration until two successive values of λ differ by less than EPSO. For subsequent sequences of λ values, a different convergence precision, XLAX, is used. After the first converged λ sequence is obtained, the initial value of the eigenvalue (EV) is altered by EVM, an input value. If $\lambda > 1$ (multiplying system), the new eigenvalue is equal to EV + EVM; if $\lambda < 1$ (decaying system), the new value is EV - EVM. These alterations correspond to the addition or the subtraction of an absorption, e.g., as in a time-absorption search or a poison-concentration search.

Basically, after two values of $k_{eff}(\lambda)$ are obtained for two different system configurations, subroutine NEWPAR attempts to fit a curve through the most recent values to extrapolate or interpolate to a value of unity. Depending on the amount of information available and the size of $|1 - \lambda|$, this fit proceeds in different ways. A parabolic fit cannot be made until three converged values of λ

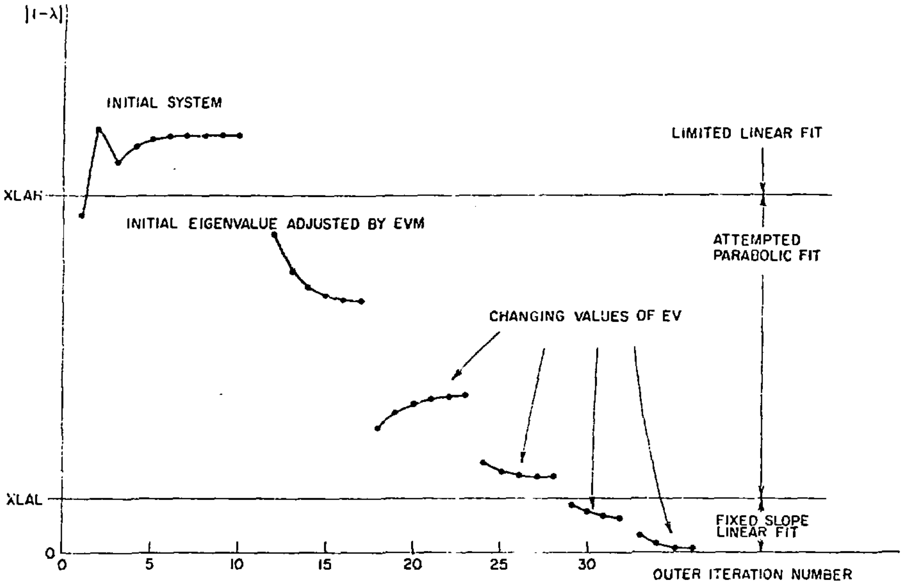


Fig. 12. Variation of λ during a hypothetical eigenvalue search.

are available, and is not attempted unless $|1 - \lambda|$ is greater than an input-search lower limit (XLAL) and less than an input-search upper limit (XLAH). If a parabolic fit is tried and the roots are imaginary, a straight-line fit is used. If the roots are not imaginary, the closest root is used as the new value of EV. Once a bracket is obtained (change of sign of $\lambda - 1$), the fit procedure is not allowed to move outside the region of the bracket. Should a parabolic fit select an eigenvalue outside the bracket region, this value is rejected and the new value is taken to be one-half the sum of the previous value and the value previous to that.

Whenever the parabolic fit is not used, a linear fit is used and the new eigenvalue is computed from

$$(EV)_{new} = (EV)_{old} + POD*EQ*(1 - \lambda) \quad , \qquad (40)$$

where POD is an input "parameter oscillation damper" which may be used to restrict the amount of change in the eigenvalue. In Eq. (40), EQ is a measure of the slope of the curve. When $|1 - \lambda| > XLAH$, $(1 - \lambda)$ in Eq. (40) is replaced by XLAH (with the correct sign) to prevent too large a change in EV. After $|1 - \lambda| < XLAL$, the value of EQ is fixed and kept constant until convergence to prevent numerical difficulty in the approximation of the derivative when λ is close to unity.

Because parametric search problems represent sequences of k_{eff} calculations, it behooves the user to study the use of subroutine NEWPAR in order to optimize his calculations. It also behooves the user to pose soluble problems. That is, there are many problems, especially concentration searches, for which solutions are not possible, and discovering this by trial and error is the hard way. Ideally, the user will have some estimate of the critical parameter available from a lower order computation.

Convergence in time-absorption calculations is typically one-sided. If EV is negative, then there is a possibility that the corrected total cross section will become negative. If this happens, the automatic search procedure may fail dramatically. For this reason POD = 0.5 or less is frequently used in such searches.

10. Adjoint Computations

The TRIPLET program solves the adjoint transport equation by transposing the matrices of scattering coefficients and inverting the group order of the problem. The solution of the resulting problem in direction $\underline{\Omega}$ is then identified with the solution of the adjoint equation in direction $-\underline{\Omega}$.[7]

The inversion of the group order is made because the transposition of the scattering matrices usually converts a downscattering problem to an upscattering problem. Because of the inversion, the user must:

(a) Enter any inhomogeneous sources, including boundary fluxes, in inverse group order,

(b) Enter any flux guess in inverse group order, and

(c) Remember that any output is in inverse group order, i.e., that groups labeled 1, 2, ..., are really groups IGM, IGM - 1, etc.

Similarly, the output flux from an adjoint problem must be inverted before insertion in a direct problem. On the other hand, an output flux from one adjoint problem is in the proper group order for use in another adjoint problem.

The group order of the group speeds and the fission spectrum is inverted by the program.

11. Edit Options

The TRIPLET user is provided with two types of edit options, zone edits and point edits. As many different zone and point edits as desired may be performed.

a. Zone Edit

An edit zone is a collection of triangles which have the same zone number. The user defines a zone by entering a set of NTC numbers (NEDZ array) which associate with each triangle on the fine mesh a zone identification number (zone i.d.). The triangles of an edit zone need not be contiguous. For each zone edit, a map of the zone i.d. similar to the material map is printed by EDMAP. For each group and zone, a table containing the zone volume, net leakage, buckling absorption, source, integral flux, average flux, and macroscopic activities (for cross-section positions 1 through IHT) is given. The macroscopic activity $A_k(g,\text{IPOS})$ in zone k and group g for cross-section position IPOS is defined by

$$A_k(g,\text{IPOS}) = \sum_i C(g,\text{IPOS},m_i)\phi_i V_i \text{ for } i \in \text{zone } k \quad ,$$

where m_i is the material i.d. (cross-section block identification number) for triangle i, $C(g,\text{IPOS},m)$ is the cross section for group g in position IPOS for material m, V_i is the triangle volume, and ϕ_i is the average flux in triangle i. Thus A_k is the activity computed with the macroscopic cross section actually used in the problem, summed over all triangles in zone k.

For each zone edit, the TRIPLET user is provided the option of calculating constituent activities and microscopic activities for any material desired. The constituent activity $A_k^j(g,\text{IPOS})$ for material j in zone k is defined by

$$A_k^j(g,\text{IPOS}) = \sum_i C(g,\text{IPOS},m_i)\phi_i V_i \delta_{jm_i} \text{ for } i \in \text{zone } k.$$

Here δ_{jm_i} equals unity if material j equals material m_i, the mixture table density (MIXDEN) if material j is a "constituent" of material m_i, and is zero otherwise. A "constituent" means that material j appears as an entry in the MIXNUM array with density MIXDEN (see Sec. III.D) that is used to form material m_i. Thus if material j is used to form a material j', which is used to form material m_i, then material j is not a "constituent" of material m_i within this definition.

The microscopic activity for material j in zone k is defined by

$$A_k^j(g,\text{IPOS}) = \sum_i C(g,\text{IPOS},j)\phi_i V_i \text{ for } i \in \text{zone } k.$$

Thus A_k^j would be the activity obtained in zone k if material j were uniformly distributed throughout the system, even though material j may not actually have appeared in the problem cross sections.

The edit input parameters NCA and NMA specify the number of constituent activities and number of microscopic activities to be calculated. The user must then enter NCA material i.d.'s for the constituent activities and NMA material i.d.'s for the microscopic activities.

To edit a material which is not actually a part of the problem, the TRIPLET user may add a mixture instruction to the mixture tables; or, if interested in only a few cross sections, he may add these cross sections to other blocks in rows IHT-5, IHT-6, etc.

Finally, following any constituent activities or microscopic activities, the zone edit provides the zone relative power density (group sum of the zone volume integral of ν × fission rate divided by the zone volume), normalized to that of a user-designated zone. The zone relative power density (unnormalized) is defined by

$$PD_k = \frac{\sum_g \sum_i C(g,IHT-1,m_i)\phi_i V_i}{\sum_i V_i} \quad \text{for } i \in \text{zone } k.$$

If the user selects zone zero (NORMZ=0), the normalization is to the whole system power density.

b. Point Edit

The point edit feature of TRIPLET provides the user with the option of obtaining the pointwise variation across each triangle of the scalar flux and activities. The user must enter the triangle i.d.'s over which the point edits are desired (NEDPT array). For each point edit, a map of the point arrangement indexing on a triangle is printed (see Fig. 9 for NP=2 example) by subroutine ARRMAP. This is followed by the pointwise scalar flux for the specified triangles. The user is also provided by the NPMA (number of point microscopic activities) parameter the option of obtaining a pointwise microscopic activity for any desired material. This pointwise microscopic activity for material j in triangle i at point k is

$$A^j_{ki}(g,IPOS) = C(g,IPOS,m_i)\phi_{ki} \quad ,$$

where ϕ_{ki} is the scalar flux at point k in triangle i.

The point edit is based on the angular flux stored on unit NAFLUX. Thus it is possible to calculate point fluxes from a calculation using the continuous model, which does not store the pointwise variation of the scalar flux.

C. Data Input Rules

Except for the control parameters, cross sections, and edit parameters, all floating-point numbers and integers are read into TRIPLET in special formats by the LOAD subroutine. These formats are [6(I1,I2,E9.4)] for reading floating-point numbers and [6(I1,I2,I9)] for integers. In each word of both of these formats, the first integer field, I1, designates the options listed below. The second integer field, I2, controls the execution of the option, and the remainder of the field, I9 or E9.4, is for the input data. All data blocks read with these formats must be ended with a 3 in the I1 field after the last word of the block. The available options are given in Table II.

TABLE II

OPTIONS FOR SPECIAL READ FORMATS

Value of I1	Nature of Option
0 or blank	No action.
1	Repeat data word in 9 field number of times indicated in I2 field.
2	Place number of linear interpolants indicated in I2 field between data word in 9 field and data word in next 9 field. Not allowed for integers.
3	Terminate reading of data block. A 3 must follow last data word of all blocks.
4	Fill remainder of block with data word in 9 field. This operation must be followed by a terminate (3).
5	Repeat data word in 9 field 10 times the value in the I2 field.
9	Skip to the next data card.

Five illustrations of the use of the special formats are given below. These illustrate:

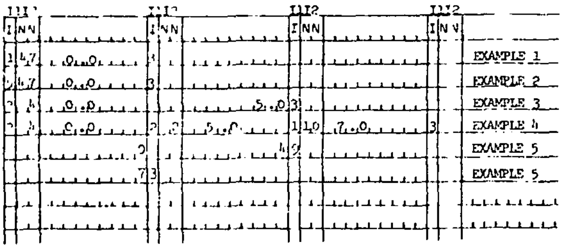
1 - Zero is repeated 47 times.

2 - Zero is repeated 470 times.

3 - Four interpolants are inserted between 0.0 and 5.0 giving six data numbers: 0.0, 1.0, 2.0, 3.0, 4.0, 5.0.

4 - Four interpolants are inserted between 0.0 and 5.0, two between 5.0 and 7.0, and 7.0 is repeated 10 times.

5 - After reading 0 and 4 we skip to the next card and read 7.



I	NN		I	NN		I	NN		I	NN	
1	47	0.0	3								EXAMPLE 1
5	47	0.0	3								EXAMPLE 2
2	4	0.0			5.0	3					EXAMPLE 3
2	4	0.0	2	2	5.0	1	10	7.0	3		EXAMPLE 4
		0			4	9					EXAMPLE 5
		7	3								EXAMPLE 5

A special routine, WRITE, is used to print some of the two- and three-dimensional arrays that occur in the program. This routine can be used for one-, two-, or three-dimensional arrays and has an option for printing a portion of an array, e.g., the mixed cross-section blocks, if any. When fluxes and fission rates are printed, they are oriented with the origin at the lower left of the page, corresponding to the picture drawn by MAPPER.

D. Description of Input Data

In the following pages the input data for TRIPLET are listed in exactly the order in which they are entered in the code. The data are divided into four categories: (1) job title cards, (2) control integers on cards 1 through 3 and control floating-point numbers on cards 4 and 5, (3) problem-dependent data on subsequent cards, and (4) edit input.

1. Job Title Cards

The user begins by indicating on a card in an I6 format the number of title or job description cards he wants to use. He then enters the descriptive material on these cards which are read with a 18A4 format.

2. Input of Control Numbers

On cards 1 through 3, the user enters the following control integers which are read in a 12I6 format and on cards 4 and 5 the following control floating-point numbers in a 6E12.4 format:

Number of Word on Card	Name of Variable	Comments
CONTROL INTEGERS (12I6)		---------- CARD 1
1	ITH	0/1 (direct/adjoint) type of calculation to be performed.
2	ISCT	0/N (isotropic/Nth-order anisotropic) order of scattering calculation. NM = (ISCT+1)(ISCT+2)/2 spherical harmonics flux components are computed. They are not used to compute a scattering source unless some cross-section material identification number is negative. See IDCS below.
3	ISN	-1/0/+1 (interface/built-in/card input) source of S_n quadrature coefficients.
4	IGM	Number of energy groups.
5	NP	Polynomial approximation order for the finite element scheme (≥ 1).
6	JT	Number of bands of triangles.
7	IBL	-3/0/1/3 (flat source/vacuum/reflective/shaped source) left boundary condition.
8	IBR	-3/0/1/2/3 (flat source/vacuum/reflective/special/shaped source) right boundary condition.
9	IBB	-3/0/1/3 (flat source/vacuum/reflective/shaped source) bottom boundary condition.
10	IBT	-3/0/1/3 (flat source/vacuum/reflective/shaped source) top boundary condition.
11	IEVT	0/1/2/3 (inhomogeneous source/k_{eff} calculation/time absorption or alpha/concentration search) eigenvalue type.
12	ISTART	-6/-5/-3/-2/-1/0/1/2/3/5/6 input flux guess and starting options. See Sec. III.B.6.b.
CONTROL INTEGERS (12I6)		---------- CARD 2
1	MM	Total number of quadrature angles in all 4 quadrants of the μ-η plane.
2	MT	Total number of materials (cross-section blocks including anisotropic cross sections) in the problem.
3	MTPS	Number of input material sets from the interface file ISOTXS. Caution: each material set from this file yields ISCT+1 materials. See IDLIB below.
4	MCR	Number of input materials from the code dependent input file. If this number is negative, FIDO format cross sections are read.
5	MS	Number of mixture instructions. See Sec. III.B.1 and items MIXNUM, MIXCOM, and MIXDEN below.

6 IHT Row of total cross section in the cross-section format. If IHT < 0, code assumes that there is no σ^{up} in the cross-section table.

7 IHS Row of within-group scattering cross section in the cross-section format.

8 IHM Total number of rows in the cross-section format.

9 IQOPT 0/1/2/3/5 options for input of inhomogeneous source. See Sec. III.B.6.a.

10 IQAN Order of anisotropy of inhomogeneous distributed source.

11 IPVT 0/1/2 (none/k_{eff}/alpha) parametric eigenvalue entered. See entry PV below.

12 IITL Maximum number of inner iterations allowed per group.

CONTROL INTEGERS (12I6) -- CARD 3

1 ISDS 0/1 (continuous/discontinuous) finite element scheme.

2 IACC 0/1 (whole system/fine mesh) rebalance acceleration method.

3 ITLIM 0/N (no/N-second) time limit. If an integer number of seconds is entered, a restart dump is taken after this number of seconds and the problem is terminated.

4 IEDOPT 0/1 (no/yes) edit input to be entered.

5 I1 0/1 (yes/no) full input flux print suppression indicator.

6 I2 0/1/2 (all/isotropic/none) final flux print indicator.

7 I3 0/1/2 (all/mixed/none) cross-section print indicator.

8 I4 0/1 (yes/no) final fission print indicator.

9 I5 0/1/2/3 (all/unnormalized/normalized/none) source print indicator.

10 ISDB 0/1 (whole system/space dependent) buckling height indicator.

11 IFO 0/1 (no/yes) interface file output is created.

CONTROL FLOATING-POINT DATA (6E12.4) -- CARD 4

1 EV Eigenvalue guess. It is satisfactory to enter 0.0 for IEVT = 2 and 1.0 for IEVT = 3.

2 EVM Eigenvalue modifier used only if IEVT > 1. See Sec. III.B.9 above.

3 PV Parametric value of k_{eff} for subcritical or supercritical systems or for 1/v absorption. See Sec. III.B.9 above.

4 XLAL Lambda lower limit for eigenvalue searches. See Sec. III.B.9 above.

5 XLAH Search lambda upper limit.

6 XLAX Search lambda convergence precision for second and subsequent values of the eigenvalue.

CONTROL FLOATING-POINT DATA (6E12.4) -- CARD 5

1 EPS Convergence precision.

2 NORM Normalization factor. Total number of source or fission particles normalized to this number if it is nonzero. No normalization if NORM is zero.

3 POD Parameter oscillation damper used in eigenvalue searches. See Sec. III.B.9.

3. Problem Dependent Data

In the input data listed below, all the items are dimensionless except for the source, flux, velocities, mesh specifications, cross sections, bucklings and mixture densities. The dimensions of these quantities are arbitrary in the following sense. Macroscopic cross sections define a unit of inverse length (usually cm^{-1} but occasionally km^{-1}) in which the mesh boundary values are measured. For source problems, the flux will have the dimensions of source/cross section where cross section is the quantity used in the calculation. Normally sources are in units of particles/length^3/solid angle/sec (the energy dependence is removed by the multigroup approximation, i.e., $\int QdE$ is used, see Sec II.B.), microscopic cross sections are in units of barns × length^2/cm^2, nuclide number densities in units of 10^{24} × number/length^3, and velocities in length/sec, although Los Alamos velocities are habitually measured in units of length/10^{-8} sec.

With the exception of the cross sections from the code dependent input file, all the following data is loaded by the LASL block loader using the special formats described in Sec. III.D. We denote these formats by S(I) for integers and S(E) for floating point numbers.

Block Name and Dimension	Format	Number of Entries	Comments
IT(JT)	S(I)	JT	Integers defining the number of triangles on each band.
ITT(JT)	S(I)	JT	1/2 (up/down) integers indicating orientation of first triangle on each band.
C(IHM,IGM,MIN)			Three options are available for reading cross sections. The LASL input format may not be mixed with the FIDO format. 1. LASL INPUT If MCR.GT.0, MCR blocks of IHM*IGM numbers are read in a 6E12.5 format. Each block is preceded by an identification card read in a 18A4 format. MIN = MCR + MTPS*(ISCT+1). 2. FIDO INPUT If MCR.LT.0, MCR blocks of data are created from FIDO input. FIDO input data must be preceded by a 14* (floating-point block number 14) loading card when an IBM 360 computer is used, but no card is required on CDC computers. 3. INTERFACE FILE ISOTXS When MTPS.GT.0, MTPS material sets are read from standard file ISOTXS. On this file each material set consists of ISCT+1 cross-section blocks for the isotropic and ISCT anisotropic cross sections. The first component of the first material is stored in cross-section block MCR+1, the first component of the second material is stored in cross-section block MCR+ISCT+2, etc. Should the ISOTXS file not contain ISCT anisotropic components, zeroes are supplied for the components not present. If the ISOTXS file contains more components than needed, only the first ISCT+1 components are read.
IDLIB(MTPS)	S(I)	MTPS	Position numbers of material sets to be read from ISOTXS. Do not enter unless MTPS.GT.0. The material sets are read in the order specified in this entry, and this order need not be in order of increasing set identification number.
Input FLUX Guess FLUX(NM,NTC,NPPT)	S(E)		Number of entries depends on option. See Sec. III.B.6.b. Option Number of entries -6 Problem restart dump from unit NDUMP1 -5 Input from RTFLUX or ATFLUX standard interface file -3 IGM, then NTC*NPPT -2 IGM sets of NTC*NPPT -1 IGM 0 None 1 NM sets of IGM 2 NM groups of (IGM, then NTC*NPPT) 3 IGM, then NM sets of NTC*NPPT

			5 Same as -5 6 Problem restart dump from unit NDUMP1
Input Source Q(NMQ,NTC,NPPT)	S(E)		Number of entries depends on option. See Sec. III.B.6.a. Option Number 0 None 1 NMQ sets of IGM 2 IGM groups of NMQ sets of NTC*NPPT 3 NMQ groups of (IGM, then NTC*NPPT)

Boundary Sources
BR(NPP,JT,MM)
BB(NPP,NUPB,MM)
BT(NPP,NDNT,MM)
BL(NPP,JT,MM)
	S(E)		The right, bottom, top, and left boundary sources (flux) are read in that order, one group at a time. For IBR, IBB, IBT, or IBL = +3, NPP = NP+1. For IBR, IBB, IBT, or IBL = -3, NPP = 1. See Sec. III.B.6.a.

WGT(MM)	S(E)	MM	S_n quadrature weights. Enter only if ISN = +1.
COSMU(MM)	S(E)	MM	S_n μ cosine directions. Enter only if ISN = +1.
COSETA(MM)	S(E)	MM	S_n η cosine directions. Enter only if ISN = +1.
HY(JT)	S(E)	JT	Height of triangle bands.
X(NTX)	S(E)	NTX	X-coordinates of triangle vertices. Entered in a single array from left to right, starting with the bottom line of the bottom band and ending with the top line of the top band.
IDCS(NTC)	S(I)	NTC	Cross-section material identification numbers. These numbers assign a cross-section block to each triangle. If these numbers are negative, an anisotropic scattering source is calculated in that triangle, but the numbers need not be negative when ISCT > 0.
CHI(IGP)	S(E)	IGM	Fission fractions. Fraction of fission yield emerging in each group.
VEL(IGP)	S(E)	IGM	Group speeds. Used only in time absorption calculations.
MIXNUM(MS)	S(I)	MS	Numbers identifying cross-section block being mixed. See Sec. III.B.1.b. Do not enter if MS = 0.
MIXCOM(MS)	S(I)	MS	Numbers controlling cross-section mixture process. See Sec. III.B.1.b. Do not enter if MS = 0.
MIXDEN(MS)	S(I)	MS	Mixture densitites. See Sec. III.B.1.b. Do not enter if MS = 0.
BHT(NTC)	S(E)	NTC or 1	Buckling heights (in cm if cross sections are in barns) used to simulate z-dimension of system by adding an absorption given by $\sigma_{a,BHT} = \frac{\sigma_t}{3} [\pi/(BHT*\sigma_t + 1.4209)]^2$. Here 1.4209 is twice the Milne problem extrapolation distance, and σ_t is the total cross section. For ISDB = 0, a single, whole-system buckling height is entered. For ISDB = 1, a buckling height is entered for each triangle. For BHT = 0, no buckling absorption is added.

4. Edit Input

The edit input, entered only if IEDOPT = 1, consists of control integers entered on cards indicated by EDIT 1, 2, or 3; and the remaining edit input entered in the special format discussed above in Sec. III.C. The zone edit control integers and the zone edit arrays are read first for all NZEDS edits, then the point edit control integers and point edit arrays are read for all NPEDS edits.

EDIT CONTROL INTEGERS (2I6) -- EDIT 1

1 NZEDS Number of zone edits.

2 NPEDS Number of point edits.

ZONE EDIT CONTROL INTEGERS, Enter only if NZEDS > 0 (4I6) -- EDIT 2

1 NZ Total number of zones.

2 NCA Number of constituent activities calculated.

3 NMA Number of microscopic activities calculated.

4 NORMZ Zone identification number for normalization of power density. If NORMZ = 0, whole system normalization is performed.

IDCA(NCA) S(I) NCA Cross-section material identification numbers for constituent activities. Enter only if NCA > 0.

IDMA(NMA) S(I) NMA Cross-section material identification numbers for microscopic activities. Enter only if NMA > 0.

NEDZ(NTC) S(I) NTC Zone identification numbers. These numbers assign a zone number to each triangle.

POINT EDIT CONTROL INTEGERS, Enter only if NPEDS > 0 (2I6) ------------------------------------- EDIT 3

1 NTPE Number of triangles to be included in the point edit.

2 NPMA Number of microscopic activities calculated in the point edit.

IDMA(NPMA) S(I) NPMA Cross-section material identification numbers for microscopic activities. Enter only if NPMA > 0.

NEDP(NTPE) S(I) NTPE Triangle identification numbers to be included in the point edit. No entries are made if NTPE = NTC; the array is automatically set to the triangle i.d. numbers.

E. Output Description for a Test Problem

The TRIPLET program comes with a set of 10 test problems. The problem output presented in the following pages is a problem designed to illustrate many of the TRIPLET options. Each page of the output is numbered, and we refer to these numbers in the text below.

As can be seen from the first output page, this problem is a 3 group, S_2, linearly anisotropic scattering, k_{eff} calculation, using a quadratic polynomial difference scheme. The integer and floating point input control data are printed on output page (1). The remaining problem input is printed on output pages (2) through (4). Note that the flux input option ISTART = -1 was used to enter the fission spectrum as an initial flux guess.

Following edit input on page (4), the table of triangle x-coordinates is printed. Each node number corresponds to an identically numbered node point on the material map. From this table and the material map, we see that the system is a 1.0 unit (here cm) by 1.0 unit square, with three reflecting boundaries and one vacuum boundary. Knowing the orientation of the first triangle on each band from the adjacent arrows, the detailed triangular mesh may be drawn by connecting successive node points on alternate node lines, as indicated.

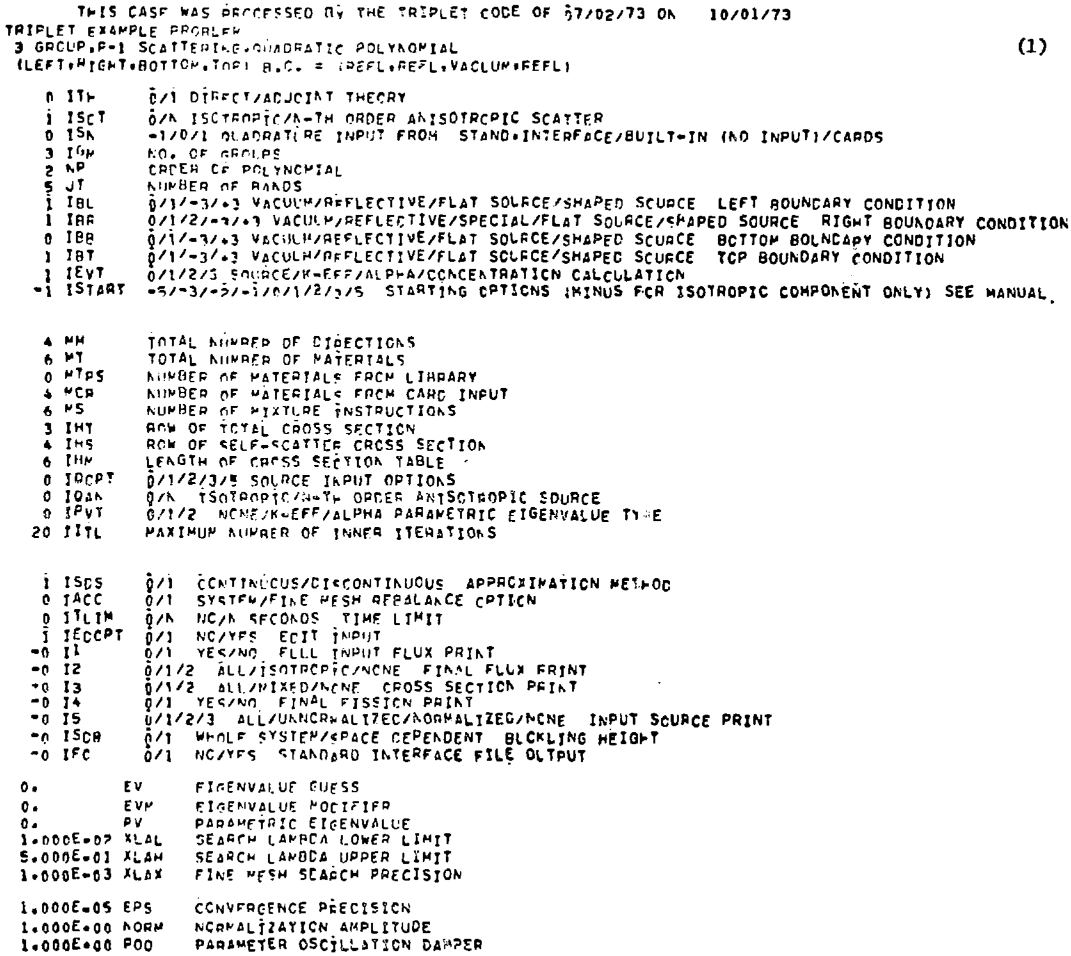
After the material map, the cross-section mixture instructions, the mixed cross sections, the fission spectrum, and the velocitites are printed.

Following the summary of convergence precisions on output page (6), a monitor of the progress of the calculation is printed. The column headed "rebalance convergence" contains the maximum deviation from one of any rebalance factor for the rebalance performed after each outer iteration. If the number of inner iterations in a group exceeds IITL, we print the maximum flux error at that time immediately to the right of the number of inner iterations.

On output page (7) the final monitor line is printed, followed by a print of the cell-averaged scalar flux and its components, for each group. The flux print starts with the top band and runs downward, with triangle 1 corresponding to the leftmost triangle in that band. The fission rate, in the same format, is the last item printed unless the edit options are exercised.

The first item of the zone edit, beginning on output page (10), is the zone map, similar to the material map, which associates each triangle with a zone number. This is followed by a table, for each group, containing for each zone the volume, net leakage, buckling absorption, total source, integral flux, average flux, and macroscopic activities. This is followed by the constituent activities for material 1, which is a constituent only of zone 2. This is followed by the microscopic activity for material 3, the activity that would be obtained if the scatterer were uniformly distributed throughout the system. The table of relative power densities appears last.

Finally, the point edit begins on output page (13). The point arrangement for both the upward and downward pointing triangles is indicated in the point arrangement map. The point flux is then printed for each triangle identification number requested for the point edit. This is then followed by the table of point microscopic activities output pages (14) through (15).



```
     THIS CASE WAS PROCESSED BY THE TRIPLET CODE OF 07/02/73 ON    10/01/73
TRIPLET EXAMPLE PROBLEM
 3 GROUP,P-1 SCATTERING,QUADRATIC POLYNOMIAL                                              (1)
 (LEFT,RIGHT,BOTTOM,TOP) B.C. = (REFL,REFL,VACUUM,REFL)

   0 ITH      0/1 DIRECT/ADJOINT THEORY
   1 ISCT     0/N ISOTROPIC/N-TH ORDER ANISOTROPIC SCATTER
   0 ISN      -1/0/1 QUADRATURE INPUT FROM  STAND.INTERFACE/BUILT-IN (NO INPUT)/CARDS
   3 IGM      NO. OF GROUPS
   2 NP       ORDER OF POLYNOMIAL
   5 JT       NUMBER OF BANDS
   1 IBL      0/1/-3/+3 VACUUM/REFLECTIVE/FLAT SOURCE/SHAPED SOURCE  LEFT BOUNDARY CONDITION
   1 IBR      0/1/2/-3/+3 VACUUM/REFLECTIVE/SPECIAL/FLAT SOURCE/SHAPED SOURCE  RIGHT BOUNDARY CONDITION
   0 IBB      0/1/-3/+3 VACUUM/REFLECTIVE/FLAT SOURCE/SHAPED SOURCE  BOTTOM BOUNDARY CONDITION
   1 IBT      0/1/-3/+3 VACUUM/REFLECTIVE/FLAT SOURCE/SHAPED SOURCE  TOP BOUNDARY CONDITION
   1 IEVT     0/1/2/3 SOURCE/K-EFF/ALPHA/CONCENTRATION CALCULATION
  -1 ISTART   -5/-3/-2/-1/0/1/2/3/5  STARTING OPTIONS (MINUS FOR ISOTROPIC COMPONENT ONLY) SEE MANUAL.

   4 MM       TOTAL NUMBER OF DIRECTIONS
   6 MT       TOTAL NUMBER OF MATERIALS
   0 MTPS     NUMBER OF MATERIALS FROM LIBRARY
   4 MCR      NUMBER OF MATERIALS FROM CARD INPUT
   6 MS       NUMBER OF MIXTURE INSTRUCTIONS
   3 IHT      ROW OF TOTAL CROSS SECTION
   4 IHS      ROW OF SELF-SCATTER CROSS SECTION
   6 IHM      LENGTH OF CROSS SECTION TABLE
   0 IQOPT    0/1/2/3/5 SOURCE INPUT OPTIONS
   0 IQAN     0/N  ISOTROPIC/N-TH ORDER ANISOTROPIC SOURCE
   0 IPVT     0/1/2  NONE/K-EFF/ALPHA PARAMETRIC EIGENVALUE TYPE
  20 IITL     MAXIMUM NUMBER OF INNER ITERATIONS

   1 ISDS     0/1  CONTINUOUS/DISCONTINUOUS  APPROXIMATION METHOD
   0 IACC     0/1  SYSTEM/FINE MESH REBALANCE OPTION
   0 ITLIM    0/N  NO/N SECONDS  TIME LIMIT
   1 IEDOPT   0/1  NO/YES  EDIT INPUT
  -0 I1       0/1  YES/NO  FULL INPUT FLUX PRINT
  -0 I2       0/1/2  ALL/ISOTROPIC/NONE  FINAL FLUX PRINT
  -0 I3       0/1/2  ALL/MIXED/NONE  CROSS SECTION PRINT
  -0 I4       0/1  YES/NO  FINAL FISSION PRINT
  -0 I5       0/1/2/3  ALL/UNNORMALIZED/NORMALIZED/NONE  INPUT SOURCE PRINT
  -0 ISCB     0/1  WHOLE SYSTEM/SPACE DEPENDENT  BUCKLING HEIGHT
  -0 IFC      0/1  NO/YES  STANDARD INTERFACE FILE OUTPUT

0.         EV      EIGENVALUE GUESS
0.         EVM     EIGENVALUE MODIFIER
0.         PV      PARAMETRIC EIGENVALUE
1.000E-02 XLAL     SEARCH LAMBDA LOWER LIMIT
5.000E-01 XLAH     SEARCH LAMBDA UPPER LIMIT
1.000E-03 XLAX     FINE MESH SEARCH PRECISION

1.000E-05 EPS      CONVERGENCE PRECISION
1.000E+00 NORM     NORMALIZATION AMPLITUDE
1.000E+00 POD      PARAMETER OSCILLATION DAMPER
```

(2)

```
INPUT TRI / BAND          5
          3               5               6               5               3


INPUT BAND TYPES          5
          2               2               2               1               1


STORAGE    REQUIRED ALLOWED
SMALL CORE    1698   41000
LARGE CORE    1981  375000
************************************************************INPUT CROSS SECTIONS**********************************************
     1                        LOADED FROM CARDS MATERIAL 1 (P=0)   FISSILE
     2                        LOADED FROM CARDS MATERIAL 1 (P=1)
     3                        LOADED FROM CARDS MATERIAL 2 (P=0)  SCATTERER
     4                        LOADED FROM CARDS MATERIAL 2 (P=1)
 INPUT CROSS SECT          1

          GROUP      1     GROUP      2     GROUP      3

    1   .100000E+01    .100000E+01    .100000E+01
    2   .250000E+01    .180000E+01   0.
    3   .260000E+01    .310000E+01    .300000E+01
    4   .100000E+01    .150000E+01    .200000E+01
    5  0.              .400000E+00    .600000E+00
    6  0.             0.              .200000E+00
 INPUT CROSS SECT          2

          GROUP      1     GROUP      2     GROUP      3

    1  0.             0.             0.
    2  0.             0.             0.
    3  0.             0.             0.
    4   .333333E+00    .222222E+00    .111111E+00
    5  0.             0.             0.
    6  0.             0.             0.
 INPUT CROSS SECT          3

          GROUP      1     GROUP      2     GROUP      3

    1   .500000E+00    .600000E+00    .200000E+00
    2  0.             0.             0.
    3   .390000E+01    .470000E+01    .420000E+01
    4   .300000E+01    .400000E+01    .420000E+01
    5  0.              .200000E+00    .100000E+00
    6  0.             0.              .200000E+00
 INPUT CROSS SECT          4

          GROUP      1     GROUP      2     GROUP      3

    1  0.             0.             0.
    2  0.             0.             0.
    3  0.             0.             0.
```

(3)

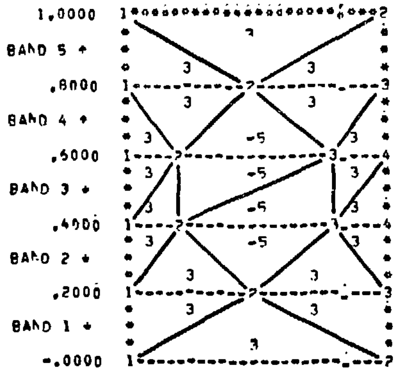
```
    4   .200000E-01  1.              0.
    5  0.            0.              0.
    6  0.            0.              0.
* * * * *  INPUT FOR FLUX   * * * * *

 ISOTROPIC COMPONENT

INPUT FLUX  ESHAPE       3
  3.0000E-01  4.0000E-01  3.0000E-01



QUADRATURE CONSTANTS   (S- 2)

                  MU              ETA              WEIGHT

      1       .57735027E+00   .57735027E+00   .25000000E+00
      2       .57735027E+00  -.57735027E+00   .25000000E+00
      3      -.57735027E+00  -.57735027E+00   .25000000E+00
      4      -.57735027E+00   .57735027E+00   .25000000E+00

INPUT BAND WIDTH        5
ALL ENTRIES =   2.0000E-01

INPUT X COORDINATES    18
  0.          1.0000E+00  0.          5.0000E-01  1.0000E+00  0.          2.0000E-01  8.0000E-01  1.0000E+00  0.
  2.0000E-01  8.0000E-01  1.0000E+00  0.          5.0000E-01  1.0000E+00  0.          1.0000E+00

INPUT CROSS SEC ID     22
           3           3           3           3           3          -5           3           3           3           3
          -5          -5           3           3           3           3          -5           3           3           3
           3           3

INPUT FISSION SPEC      3
  3.0000E-01  4.0000E-01  3.0000E-01

INPUT VELOCITIES        3
ALL ENTRIES =   1.0000E+00

INPUT MIX NUMBERS       6
           5           5           5           6           6           6

INPUT MIX COMMANDS      6
           0           1           3           0           2           4

INPUT MIX DENSITY       6
  0.          6.0000E-01  4.0000E-01  0.          6.0000E-01  4.0000E-01

INPUT BUCKLING WTS      1
ALL ENTRIES =   0.


EDIT INPUT

   1  NUMBER OF ZONE EDITS
   1  NUMBER OF POINT EDITS
```

(4)

```
ZONE EDIT NUMBER   1

   2  NUMBER OF ZONES
   1  NUMBER OF CONSTITUENT ACTIVITIES
   1  NUMBER OF MICROSCOPIC ACTIVITIES
   2  ZONE NORMALIZATION ID


INPUT CONST.ACT ID      1
ALL ENTRIES =             1

INPUT MICRO.ACT ID      1
ALL ENTRIES =             3

INPUT ZONE IDS,        22
          1            1            1            1            1            2            1            1            1            1
          2            2            1            1            1            1            2            1            1            1
          1            1


 POINT EDIT NUMBER   1

   4  NUMBER OF TRIANGLES
   1  NUMBER OF MICROSCOPIC ACTIVITIES


INPUT MICRO.ACT ID      1
ALL ENTRIES =             3

INPUT TRIANGLE IDS      4
          6            11           12           17

TABLE OF TRIANGLE X-COORDINATES
    BOTTOM NODES OF BAND TO LEFT,TOP NODES OF BAND TO RIGHT

NODE            BAND   1       BAND   2       BAND   3       BAND   4       BAND   5
   1       0.             0.             0.             0.             0.             0.
   2       1.0000E+00     5.0000E-01     2.0000E-01     2.0000E-01     5.0000E-01     1.0000E+00
   3                      1.0000E+00     8.0000E-01     8.0000E-01     1.0000E+00
   4                                     1.0000E+00     1.0000E+00
```

```
******************************************** FINE MESH MATERIAL MAP ********************************************
    DIRECTION IN WHICH FIRST TRIANGLE POINTS    +=UP  +=DOWN
    BOUNDARY CONDITION CHARACTERS    (-=VACUUM OR SOURCE) (*=REFLECTING) (.=SPECIAL)
    LEFT NUMBER INDICATES Y-COORDINATE
```

(5)



```
$$$$$$$$$$$$$$$$$$$$$ INPUT COMPLETED $$$$$$$$$$$$$$$$$$$$$




MIXTURE NUMBER MIXTURE COMMAND  MATERIAL ATOMIC DENSITY
         5                0              0.                               1
         5                1              6.0000000E-01                    2
         5                3              4.0000000E-01                    3
         6                0              0.                               4
         6                2              6.0000000E-01                    5
         6                4              4.0000000E-01                    6

GROUP NUMBER    1

MIXED X-SECT

       MATERL    5   MATERL    6

    1   .800000E+00  0.
    2   .150000E+01  0.
    3   .312000E+01  0.
    4   .180000E+01   .208000E+00
    5  0.            0.
    6  0.            0.

GROUP NUMBER   2

MIXED X-SECT
```

(6)

```
          MATERL     5    MATERL     6

     1   .940000E+00  0.
     2   .108000E+01  0.
     3   .374000E+01  0.
     4   .250000E+01  .133333E+00
     5   .320000E+00  0.
     6  0.            0.

GROUP NUMBER   3

MIXED X-SECT

          MATERL     5    MATERL     6

     1   .680000E+00  0.
     2  0.            0.
     3   .348000E+01  0.
     4   .290000E+01  .666666E-01
     5   .400000E+00  0.
     6   .200000E+00  0.

GROUP FISSION SPECTRUM     VELOCITIES

  1       3.0000000E-01   1.0000000E+00
  2       4.0000000E-01   1.0000000E+00
  3       3.0000000E-01   1.0000000E+00
  4       1.0000000E+00   0.

EPSO   1.00E-05       XLAX   1.00E-03
EPSI   1.00E-05       XLAH   5.00E-01
EPSR   0.             XLAL   1.00E-02
EPSX   1.00E-05       PCD    1.00E+00
NORM   1.00E+00

IITL   19


TIME IN     OUTER        INNER          NEUTRON        EIGENVALUE      EIGENVALUE      LAMBDA          REBALANCE
MINUTES ITERATIONS  ITERATIONS          BALANCE                          SLOPE                         CONVERGENCE
                    TOTAL/BY GROUP

 6.64E-03    0        0                0.             0.              0.              0.              0.
   3.77E-02                    1   27  5.84E-05
   3.82E-02                    2   2*  3.82E-04
   3.90E-02                    3   2*  1.18E-03
 1.21E-01    1        40               1.52923561E-05 3.27820349E-01 0.               3.27765642E-01 1.66909838E-04
   2.93E-02                    1   15
   3.50E-02                    2   19
   3.78E-02                    3   20  2.15E-05
 2.28E-01    2        54               2.90126536E-07 3.31263043E-01 0.               1.01849857E+00 3.16793923E-06
   2.45E-02                    1   17
   3.02E-02                    2   16
   3.02E-02                    3   16
 3.15E-01    3        45               2.22418414E-07 3.32295703E-01 0.               1.08311734E+00 2.42862675E-06
   2.08E-02                    1   11
   2.46E-02                    2   13
  2.50E-02                        3  12
3.83E-01      4        37                 2.10326930E-07 3.32609942E-01 0.                  1.00094566E+00 2.29671691E-06
  1.70E-02                        1   9                                                                                         (7)
  2.09E-02                        2  11
  1.90E-02                        3  10
4.49E-01      5        30                 1.88777705E-07 3.32707650E-01 0.                  1.00029376E+00 2.06148953E-06
  1.33E-02                        1   7
  1.53E-02                        2   8
  1.15E-02                        3   6
4.95E-01      6        21                 2.22975672E-07 3.32738499E-01 0.                  1.00009272E+00 2.43499036E-06
  9.52E-03                        1   5
  9.72E-03                        2   5
  9.54E-03                        3   5
5.22E-01      7        15                 1.94576749E-07 3.32748276E-01 0.                  1.00002938E+00 2.12489070E-06
  7.62E-03                        1   4
  7.58E-03                        2   4
  5.72E-03                        3   3
5.53E-01      8        11                 1.56004887E-07 3.32751670E-01 0.                  1.00001020E+00 1.70367478E-06
  3.82E-03                        1   2
  3.87E-03                        2   2
  1.94E-03                        3   1
5.68E-01      9         5                 1.71024041E-07 3.32752657E-01 0.                  1.00000297E+00 1.86769719E-06
*************************************************************************************************************************

*FINAL*DUMP TAKEN ON UNIT        7

*************************************************************************************************************************




FINAL PRINTING




TIME IN     OUTER       INNER          NEUTRON          EIGENVALUE       EIGENVALUE        LAMBDA          REBALANCE
MINUTES ITERATIONS  ITERATIONS         BALANCE                             SLOPE                           CONVERGENCE
                    TOTAL/BY GROUP

 5.84E-01      10        6            3.26179716E-07 3.32752984E-01 0.                  1.00000098E+00 1.86769719E-06


FLUX COMPONENTS FOR GROUP  1
ISOTROPIC COMPONENT
BAND      TRIANGLE    1   TRIANGLE    2   TRIANGLE    3   TRIANGLE    4   TRIANGLE    5   TRIANGLE    6
  5          2.30364E-01     2.0213 E-01     2.29993E-01
  4          1.97350E-01     2.55064E-01     4.22125E-01     2.54639E-01     1.97406E-01
  3          1.57848E-01     2.12028E-01     4.40893E-01     4.30414E-01     2.16169E-01     1.54152E-01
  2          1.70028E-01     2.07143E-01     3.77109E-01     2.07442E-01     1.69648E-01
```

(8)

```
   1        1.39193E-01    1.00387E-01    1.39473E-01
COMPONENT 2
BAND       TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5        -1.63551E-02   6.63425E-05    1.62195E-02
   4        -1.96265E-02  -3.90574E-02    1.09573E-05    3.91551E-02    1.96930E-02
   3        -1.60523E-02  -4.62831E-02   -3.14472E-02    3.13974E-02    4.53743E-02    1.71541E-02
   2        -2.13460E-02  -4.13597E-02    1.84579E-04    4.12394E-02    2.10948E-02
   1        -1.75496E-02  -3.49276E-05    1.78510E-02
COMPONENT 3
BAND       TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5         2.13275E-02   1.63033E-02    2.12427E-02
   4         1.72599E-02   4.15153E-02    6.12259E-02    4.13952E-02    1.73023E-02
   3         1.84990E-03  -1.81681E-02    2.04291E-03   -2.85355E-02    2.37942E-04   -1.67765E-02
   2        -3.66414E-02  -6.14974E-02   -8.24299E-02   -6.13654E-02   -3.65704E-02
   1        -5.14763E-02  -4.48193E-02   -5.15910E-02


FLLX COMPCNENTS FOR GRCUP  2
ISCTROPIC COMPONENT
BAND       TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5         4.26961E-01   4.00361E-01    4.26356E-01
   4         3.89242E-01   4.75169E-01    7.18682E-01    4.73436E-01    3.89315E-01
   3         3.39478E-01   4.08064E-01    7.42038E-01    7.22695E-01    4.19192E-01    3.28771E-01
   2         3.24758E-01   3.72397E-01    6.31462E-01    3.72854E-01    3.23443E-01
   1         2.39649E-01   1.78231E-01    2.40656E-01
COMPONENT 2
BAND       TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5        -2.07131E-02   5.46203E-05    2.05491E-02
   4        -2.56246E-02  -5.79634E-02    3.78638E-05    5.03386E-02    2.56473E-02
   3        -2.31235E-02  -6.22437E-02   -4.19788E-02    4.17398E-02    6.12518E-02    2.43156E-02
   2        -2.72386E-02  -5.25556E-02    2.06846E-04    5.24143E-02    2.70787E-02
   1        -2.13717E-02  -5.27351E-05    2.17513E-02
COMPONENT 3
BAND       TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5         2.64433E-02   2.15711E-02    2.63313E-02
   4         1.77443E-02   5.18549E-02    7.76426E-02    5.16941E-02    1.78121E-02
   3        -2.57640E-03  -2.77134E-02   -9.20493E-04   -4.23061E-02   -4.50130E-03   -2.59660E-02
   2        -5.20996E-02  -8.77482E-02   -1.15365E-01   -8.73232E-02   -5.20165E-02
   1        -7.55291E-02  -7.23243E-02   -7.56708E-02


FLLX COMPCNENTS FOR GRCUP  3
ISCTROPIC COMPONENT
BAND       TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5         1.03131E+00   1.00478E+00    1.03036E+00
   4         9.38580E-01   1.03276E+00    1.20990E+00    1.03182E+00    9.37827E-01
   3         8.34449E-01   8.55307E-01    1.17531E+00    1.12295E+00    8.99308E-01    7.91109E-01
   2         7.08115E-01   7.36649E-01    9.70921E-01    7.07148E-01    7.07469E-01
   1         4.85349E-01   3.47044E-01    4.85823E-01
COMPONENT 2
BAND       TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5        -1.69343E-02   1.40283E-04    1.67107E-02
   4        -2.19989E-02  -4.25027E-02    1.42544E-05    4.25116E-02    2.19831E-02
   3        -2.04881E-02  -5.30154E-02   -3.59242E-02    3.59707E-02    5.14005E-02    2.21908E-02
   2        -2.51543E-02  -4.88521E-02    2.18642E-04    4.87612E-02    2.49906E-02
   1        -2.10099E-02  -5.86381E-05    2.14548E-02
COMPONENT 3
BAND       TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5         1.11956E-02   1.29898E-02    1.10634E-02
```

```
    4       -1.43623E-02    1.89644E-02    3.38007E-02    1.88306E-02   -1.43729E-02
    3       -4.26477E-02   -6.91951E-02   -4.33993E-02   -8.55423E-02   -4.46981E-02
    2       -1.02996E-01   -1.41372E-01   -1.60849E-01   -1.41436E-01   -1.02937E-01   -6.71968E-02
    1       -1.44755E-01   -1.45373E-01   -1.44911E-01
```

(9)

```
FISSION EDIT


BAND        TRIANGLE   1   TRIANGLE   2   TRIANGLE   3   TRIANGLE   4   TRIANGLE   5   TRIANGLE   6
   5          0.              0.             0.
   4          0.              0.             1.40936E+00    0.             0.
   3          0.              0.             1.46274E+00    1.42614E+00    0.             0.
   2          0.              0.             1.24764E+00    0.             0.
   1          0.              0.             0.
```

```
*************************************************************  ZONE EDIT NUMBER    1*******************************************************************

SEE MATERIAL MAP FOR TABLE OF TRIANGLE X-COORDINATES
    DIRECTION IN WHICH FIRST TRIANGLE POINTS   +=UP  +=DOWN
    BOUNDARY CONDITION CHARACTERS   (-=VACUUM OR SOURCE) (*=REFLECTING) (.=SPECIAL)
    LEFT NUMBER INDICATES Y-COORDINATE
```

(10)

```
   1.0000  1*********************.****2
           *           1             *
 BAND 5 +  *                         *
           *     1           1       *
    .8000  1-----------2----------.---3
           *     1           1       *
 BAND 4 +  *                         *
           * 1         2          1  *
    .6000  1----2---------------3.---4
           * 1         2          1  *
 BAND 3 +  *                         *
           * 1         2          1  *
    .4000  1----2---------------3.---4
           * 1         2          1  *
 BAND 2 +  *                         *
           *     1           1       *
    .2000  1-----------2----------.---3
           *     1           1       *
 BAND 1 +  *                         *
           *           1             *
   -.0000  1----------------------.---2


                               GROUP   1

  ZONE          VOLUME          NET LEAKAGE       BUCKLING ABS           SOURCE          INTEGRAL FLUX       AVERAGE FLUX
   1          .7600000E+00     -.1286194E+00      0.               0.                     .1429099E+00       .1880394E+00
   2          .2400000E+00      .1676931E+00      0.               0.                     .1002324E+00       .4176351E+00
MACROSCOPIC ACTIVITIES

X.SEC.
POS      ZONE   1          ZONE   2
  1    7.145496E-02     8.016595E-02
  2    0.               1.503487E-01
  3    5.573487E-01     3.127252E-01
                               GROUP   2

  ZONE          VOLUME          NET LEAKAGE       BUCKLING ABS           SOURCE          INTEGRAL FLUX       AVERAGE FLUX
   1          .7600000E+00     -.1889670E+00      0.               0.                     .2677828E+00       .3523458E+00
   2          .2400000E+00      .2226470E+00      0.               0.                     .1688929E+00       .7037204E+00
MACROSCOPIC ACTIVITIES

X.SEC.
POS      ZONE   1          ZONE   2
  1    1.606657E-01     1.418703E-01
  2    0.               1.824043E-01
  3    1.258579E+00     6.316594E-01
                               GROUP   3

  ZONE          VOLUME          NET LEAKAGE       BUCKLING ABS           SOURCE          INTEGRAL FLUX       AVERAGE FLUX
```

```
    1          .7600000E+00    -.6388117E-01    0.              0.                   .5961917E+00    .7844628E+00
    2          .2400000E+00     .2054442E+00    0.              0.                   .2678808E+00    .1116170E+01
MACROSCOPIC ACTIVITIES                                                                                                  (11)

X.SEC.
POS       ZONE   1        ZONE   2
  1    1.197383E-01    1.821599E-01
  2    0.              0.
  3    2.504005E+00    9.322252E-01


                               ALL GROUPS

  ZONE             VOLUME          NET LEAKAGE      BUCKLING ABS          SOURCE        INTEGRAL FLUX     AVERAGE FLUX
   1           .7600000E+00    -.3513676E+00    0.              0.                   .1006884E+01    .1324848E+01
   2           .2400000E+00     .5957844E+00    0.              0.                   .5370061E+00    .2237526E+01
MACROSCOPIC ACTIVITIES

X.SEC.
POS       ZONE   1        ZONE   2
  1    3.513630E-01    4.042149E-01
  2    0.              3.327530E-01
  3    4.319933E+00    1.876610E+00



- - - - - - - - - - - - - - - - - - - - - - - - CONSTITUENT ACTIVITIES FOR MATERIAL   1 - - - - - - - - - - - - - - - - - - - - - - -
                               GROUP   1
X.SEC.
POS       ZONE   1        ZONE   2
  1    0.              6.013946E-02
  2    0.              1.503487E-01
  3    0.              1.563626E-01
                               GROUP   2
X.SEC.
POS       ZONE   1        ZONE   2
  1    0.              1.012357E-01
  2    0.              1.824043E-01
  3    0.              3.141408E-01
                               GROUP   3
X.SEC.
POS       ZONE   1        ZONE   2
  1    0.              1.607285E-01
  2    0.              0.
  3    0.              4.921855E-01


                               ALL GROUPS
X.SEC.
POS       ZONE   1        ZONE   2
  1    0.              3.222037E-01
  2    0.              3.327530E-01
  3    0.                9.526088E-01
```

(12)

```
- - - - - - - - - - - - - - - - - - - - - - - MICROSCOPIC ACTIVITIES FOR MATERIAL    3 - - - - - - - - - - - - - - - - - - - - - - -
                                   GROUP    1
X.SEC.
POS       ZONE    1          ZONE    2
  1    7.145496E-02      5.011622E-02
  2    0.                0.
  3    5.573487E-01      3.909065E-01
                                   GROUP    2
X.SEC.
POS       ZONE    1          ZONE    2
  1    1.606697E-01      1.013357E-01
  2    0.                0.
  3    1.259579E+00      7.937966E-01
                                   GROUP    3
X.SEC.
POS       ZONE    1          ZONE    2
  1    1.192383E-01      5.357616E-02
  2    0.                0.
  3    2.504005E+00      1.125039E+00


                                ALL GROUPS
X.SEC.
POS       ZONE    1          ZONE    2
  1    3.513630E-01      2.050281E-01
  2    0.                0.
  3    4.319933E+00      2.309803E+00



                                    RELATIVE
                                    POWER
                              ZONE  DENSITY
                                1   0.
                                2   1.000000E+00
```

(13)

```
************************************************************  POINT EDIT NUMBER  1  ************************************************************
POINT ARRANGEMENT ON THE TRIANGLE IS AS SHOWN

         *             *
         6        3    2    1
       4   5          5   4
     1   2   3          6


                                P O I N T      F L U X       F O R       G R O U P     1
PT       TRI.    6         TRI.    11          TRI.    12         TRI.    17
 1     2.451478E-01      2.614519E-01       2.425495E-01      2.611108E-01
 2     4.799335E-01      5.099885E-01       4.769400E-01      5.123943E-01
 3     2.446129E-01      2.692087E-01       2.500769E-01      2.590769E-01
 4     3.281393E-01      5.245642E-01       5.243359E-01      3.786522E-01
 5     3.233536E-01      2.890264E-01       2.899668E-01      3.753269E-01
 6     2.657212E-01      2.446607E-01       2.666736E-01      3.392222E-01

                                P O I N T      F L U X       F O R       G R O U P     2
PT       TRI.    6         TRI.    11          TRI.    12         TRI.    17
 1     4.478790E-01      4.839474E-01       4.429028E-01      4.845845E-01
 2     7.896705E-01      8.446155E-01       7.857525E-01      8.493232E-01
 3     4.467225E-01      4.940850E-01       4.541343E-01      4.811900E-01
 4     5.557887E-01      8.631372E-01       8.627205E-01      6.559131E-01
 5     5.489275E-01      5.183625E-01       5.196255E-01      6.508085E-01
 6     4.463113E-01      4.447977E-01       4.906356E-01      5.961964E-01

                                P O I N T      F L U X       F O R       G R O U P     3
PT       TRI.    6         TRI.    11          TRI.    12         TRI.    17
 1     8.447129E-01      9.683107E-01       8.391676E-01      9.708637E-01
 2     1.150814E+00      1.205605E+00       1.147296E+00      1.294598E+00
 3     8.437898E-01      9.797799E-01       8.505478E-01      9.677488E-01
 4     8.845923E-01      1.266265E+00       1.265976E+00      1.156928E+00
 5     8.773563E-01      9.550745E-01       9.555735E-01      1.150563E+00
 6     7.150814E-01      8.402445E-01       9.731599E-01      1.123046E+00


* * * * * * * * * * * * * * * * * * * * * * * * MICROSCOPIC ACTIVITIES FOR MATERIAL     3 * * * * * * * * * * * * * * * * * * * * * * *

                                                                       GROUP    1

CROSS SECTION TABLE POSITION  1
PT       TRI.    6         TRI.    11          TRI.    12         TRI.    17
 1     1.225739E-01      1.307259E-01       1.212748E-01      1.305554E-01
 2     2.399168E-01      2.545462E-01       2.384706E-01      2.561972E-01
 3     1.223065E-01      1.341044E-01       1.250385E-01      1.295385E-01
 4     1.640697E-01      2.622821E-01       2.621680E-01      1.893261E-01
 5     1.616760E-01      1.445132E-01       1.449834E-01      1.876635E-01
 6     1.328606E-01      1.220304E-01       1.333368E-01      1.696111E-01
CROSS SECTION TABLE POSITION  2   ALL ACTIVITIES ZERO

CROSS SECTION TABLE POSITION  3
PT       TRI.    6         TRI.    11          TRI.    12         TRI.    17
 1     9.560764E-01      1.019662E+00       9.459432E-01      1.018332E+00
 2     1.871351E+00      1.985445E+00       1.860066E+00      1.998338E+00
```

(14)

```
  3     9.539904E-01     1.046014E+00     9.753000E-01     1.010400E+00
  4     1.279743E+00     2.045800E+00     2.044910E+00     1.476744E+00
  5     1.261079E+00     1.127203E+00     1.130870E+00     1.463775E+00
  6     1.036313E+00     9.518368E-01     1.040027E+00     1.322967E+00

                                                                   GROUP   2

CROSS SECTION TABLE POSITION  1
PT      TRI.    6        TRI.    11        TRI.    12        TRI.    17
  1     2.687274E-01     2.901284E-01     2.657417E-01     2.907507E-01
  2     4.738023E-01     5.067693E-01     4.714515E-01     5.095939E-01
  3     2.680335E-01     2.964534E-01     2.724806E-01     2.887140E-01
  4     3.324702E-01     5.178823E-01     5.176323E-01     3.935478E-01
  5     3.293565E-01     3.110175E-01     3.117753E-01     3.904851E-01
  6     2.677868E-01     2.668786E-01     2.943813E-01     3.577179E-01
CROSS SECTION TABLE POSITION  2    ALL ACTIVITIES ZERO

CROSS SECTION TABLE POSITION  3
PT      TRI.    6        TRI.    11        TRI.    12        TRI.    17
  1     2.105031E+00     2.272673E+00     2.081643E+00     2.277547E+00
  2     3.711451E+00     3.969693E+00     3.693037E+00     3.991819E+00
  3     2.059596E+00     2.322218E+00     2.134431E+00     2.261593E+00
  4     2.612207E+00     4.056745E+00     4.054786E+00     3.082791E+00
  5     2.579959E+00     2.436304E+00     2.442240E+00     3.058800E+00
  6     2.097663E+00     2.090549E+00     2.305987E+00     2.802123E+00

                                                                   GROUP   3

CROSS SECTION TABLE POSITION  1
PT      TRI.    6        TRI.    11        TRI.    12        TRI.    17
  1     1.689426E-01     1.936637E-01     1.678334E-01     1.941727E-01
  2     2.301628E-01     2.579210E-01     2.294591E-01     2.588195E-01
  3     1.687578E-01     1.959560E-01     1.701096E-01     1.935498E-01
  4     1.769185E-01     2.532520E-01     2.531951E-01     2.313656E-01
  5     1.754713E-01     1.910149E-01     1.911147E-01     2.301127E-01
  6     1.430163E-01     1.696490E-01     1.946318E-01     2.246093E-01
CROSS SECTION TABLE POSITION  2    ALL ACTIVITIES ZERO

CROSS SECTION TABLE POSITION  3
PT      TRI.    6        TRI.    11        TRI.    12        TRI.    17
  1     3.547794E+00     4.046939E+00     3.524501E+00     4.077628E+00
  2     4.833419E+00     5.416340E+00     4.818642E+00     5.435210E+00
  3     3.543913E+00     4.115075E+00     3.572301E+00     4.064545E+00
  4     3.715288E+00     5.318312E+00     5.317098E+00     4.858677E+00
  5     3.694896E+00     4.011313E+00     4.013409E+00     4.832367E+00
  6     3.003342E+00     3.529027E+00     4.007267E+00     4.716795E+00


                                                                ALL GROUPS

CROSS SECTION TABLE POSITION  1
PT      TRI.    6        TRI.    11        TRI.    12        TRI.    17
  1     5.602439E-01     6.145181E-01     5.548499E-01     6.154788E-01
  2     9.438819E-01     1.019234E+00     9.393806E-01     1.024611E+00
  3     5.590977E-01     6.265137E-01     5.676286E-01     6.118022E-01
  4     6.744613E-01     1.033417E+00     1.032995E+00     8.142395E-01
```

```
 5     6.665045E-01     6.465456E-01     6.478734E-01     8.082612E-01
 6     5.436637E-01     5.569579E-01     6.223499E-01     7.519382E-01
CROSS SECTION TABLE POSITION  2   ALL ACTIVITIES ZERO

CROSS SECTION TABLE POSITION  3
PT      TRI.    6       TRI.    11       TRI.    12       TRI.    17
 1     6.608902E+00     7.759274E+00     6.552088E+00     7.373507E+00
 2   . 1.041622E+01     1.137148E+01     1.037174E+01     1.142537E+01
 3     6.557499E+00     7.483308E+00     6.682032E+00     7.336533E+00
 4     7.697238E+00     1.142086E+01     1.141679E+01     9.418212E+00
 5     7.525934E+00     7.574820E+00     7.586519E+00     9.354941E+00
 6     6.137318E+00     6.571413E+00     7.433281E+00     8.841885E+00



***** TOTAL EXECUTION TIME IN MINUTES =   5.8815E-01
```

IV. PROGRAMMING INFORMATION

In this section we give some of the details of the TRIPLET program. The material contained in this section is designed to help in the future local modification of the program. Much supplementary information is provided by the program comment cards.

A. Program Structure

1. Role and Function of Subprograms

We describe in Table IV the function of all the subprograms in TRIPLET. In Table IV, we follow the overlay structure defined in Table III. The overlay is of the simple control, input, execute and output kind. Note that on IBM machines with large central memories it is more efficient to remove the substructure of OVERLAY(2,0).

2. Program Flow

In Fig. 13 we show an amplified version of Fig. 7, depicting the overall flow of the TRIPLET program and showing subroutine names.

3. Relation of Problem Variables and Program Mnemonics

In much of the material in this manual we have used variables actually appearing in the FORTRAN of the program. A list of the relations between problem variable symbols and program variable names is given in Table V.

4. Definition of Variables in Common Blocks

Tables VI through XI define the variables stored in blank common block IA and the named common blocks of TRIPLET. The container array A, for problem data is also in blank common. Block IA contains problem input parameters, first word addresses of data stored in the A array, and data generated by the program. Block FWBGNI contains information necessary to a problem restart.

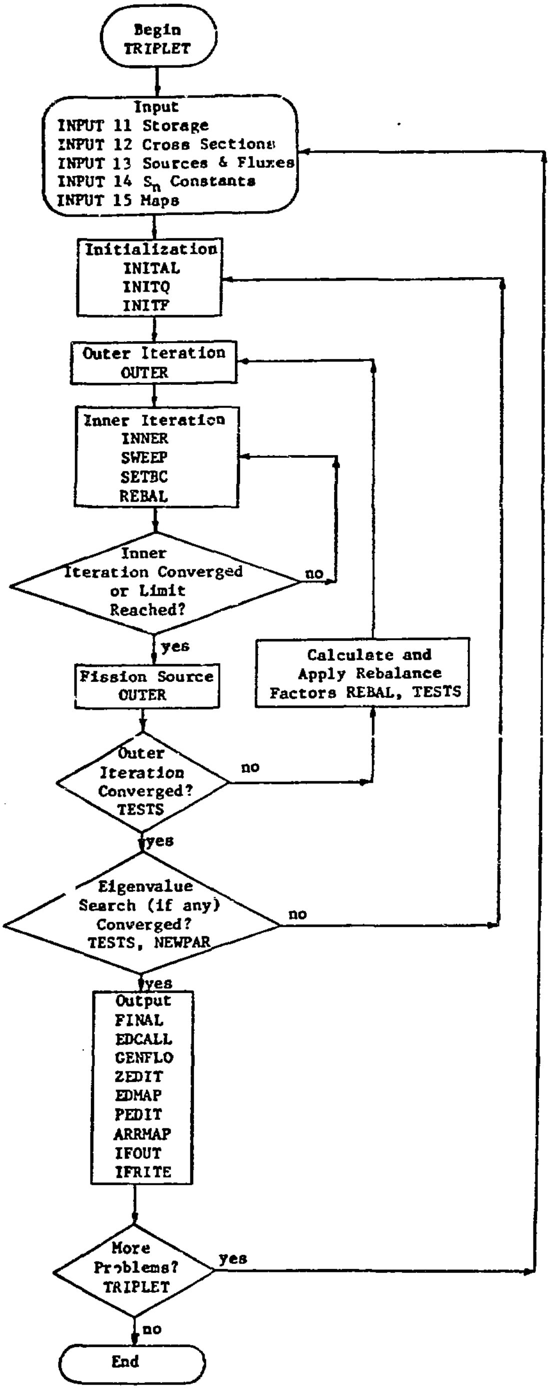


Fig. 13. Amplified form of flow diagram in Fig. 7 showing subroutine names in overall program logic.

TABLE III

STRUCTURE OF THE TRIPLET PROGRAM[a]

Overlay (0,0)
TRIPLET

1. MONITR
2. ERROR
3. CLEAR
4. MPLY
5. WRITE
6. ECHECK
7. DUMPER
8. REED
9. RITE
10. LSS
11. DOTPRO

Overlay (1,0)
INPUT1

1. LOAD
2. INPUT11
   - a. DUMPRD
   - b. KSET
3. INPUT12
   - a. CSPREP
   - b. IFINXS
4. INPUT13
   - a. READQF
   - b. IFINQF
5. INPUT14
   - a. SNCON
   - b. IFINSN
   - c. PNGEN
6. INPUT15
   - a. IXCHK
   - b. BCCHK
   - c. HYCHK
   - d. POLY
   - e. PMULTI
   - f. PINT1
   - g. PINT2
   - h. PMULT
   - i. PINT
   - j. FACT
   - k. SIAF
   - l. BANDID
   - m. MAPPER
   - n. CONVT

Overlay (2,0)
GRIND2

1. REBAL
2. GRIND21
   - a. INITAL
   - b. INITQ
   - c. INITF
3. GRIND22
   - a. OUTER
   - b. INNER
   - c. NEWAB
   - d. SWEEP
   - e. SETBC
4. GRIND23
   - a. TESTS
   - b. NEWPAR

Overlay (3,0)
OUTPUT3

1. OUTPT31
   - a. FINAL
2. OUTPUT32
   - a. EDCALL
   - b. GENFLO
   - c. ZEDIT
   - d. BANDED
   - e. EDMAP
   - f. CONTV
   - g. PEDIT
   - h. ARRMAP
   - i. CONVRT
3. IFOUT
   - a. IFRITE

[a] On IBM 360 with large bulk core storage, it is computationally more efficient to remove the substructure of OVERLAY(2,0), locating GRIND21, GRIND22, and GRIND23 in GRIND2. For the IBM version, variable names of more than seven characters are shortened to six. The underlined names are overlay segment names.

TABLE IV

FUNCTION OF TRIPLET SUBROUTINES

(Overlay) Subroutine	Function
(0,0) TRIPLET	Main driver of program. Initializes program parameters; calls input, calculation and output overlays; and attempts to execute sequential problems.
1. MONITR	Prints resumé of convergence parameters, monitor line headings, and outer iteration monitor data.
2. ERROR	Prints messages, including error notices.
3. CLEAR	Stores a constant in an array.
4. MPLY	Multiplies an array by a constant.
5. WRITE	Generalized output routine for printing 1D, 2D or 3D arrays, either integer or floating point.
6. ECHECK	Checks equality of elements in a vector. Used to reduce printing in WRITE.
7. DUMPER	Writes a restart dump.
8. REED	Handles all binary reading operations including rewind and bulk memory transfers (ECS or LCM).
9. RITE	Handles all binary writing operations including end of file and rewind and bulk memory transfers.
10. LSS	Linear system of equations solver. Used in POLY and to solve for fluxes on a triangle in SWEEP.
11. DOTPRO	Calculates dot product. Used by LSS.
(1,0) INPUT1	Calls five sections of this overlay.
1. LOAD	Los Alamos data loader.
(1,1) 2. INPUT11	Reads basic parameter data, checks it, calculates storage locations in the container array and in bulk storage, prints basic parameter data. Calls DUMPRD.
a. DUMPRD	Reads restart dump.
b. KSET	Calculates integers NTC, NTX, NUPB, NDNT, and indexing arrays KC and KX.
(1,2) 3. INPUT12	Clears cross-section, flux, and flow bulk storage. Calls CSPREP.
a. CSPREP	Reads cross sections in standard LASL format, FIDO format or from interface file by calling IFINXS. Prints cross sections, performs adjoint transpositions and reversals, checks cross sections, and calculates effective absorptions.
b. IFINXS	Interface input of cross sections from standard file ISOTXS.
(1,3) 4. INPUT13	Calls READQF to read source or flux guess.
a. READQF	Reads source or flux guess from cards or calls IFINQF to read source or flux guess from interface files. Prints flux if required.
b. IFINQF	Reads source or flux from standard interface file.
(1,4) 5. INPUT14	Reads S_n constants and generates spherical harmonic functions.
a. SNCON	Provides built-in library of S_n constants, S_2 through S_{16}.
b. IFINSN	Interface input of S_n constants from standard file SNCONS.
c. PNGEN	Generates polynomials $R_n^k(\mu,\phi)$.
(1,5) 6. INPUT15	Reads data specified by input parameters, including edit information. Calls POLY and MAPPER.
a. IXCHK	Checks that x-coordinates form a monotonically increasing sequence on each band.
b. BCCHK	Checks that left or right boundaries are vertical if they are reflecting or special.

c. HYCHK Checks that band heights are symmetric about center line if special boundary condition is used.

d. POLY Generates arrays of inner products used to solve for angular fluxes on a standard right triangle in SWEEP.

e. PMULT1 Multiplies two one-dimensional polynomials together.

f. PINT1 Integrates a one-dimensional polynomial over the interval (0,1).

g. PINT2 Integrates a one-dimensional polynomial over the interval $(0,\sqrt{2})$.

h. PMULT Multiplies two two-dimensional polynomials together.

i. PINT Integrates a two-dimensional polynomial over the standard right triangle.

j. FACT Calculates factorial function, n!

k. SIAF Generates IAF array used for indexing points on a triangle during rotations.

l. BANDID Generates IBAND array, used in REBAL and MAPPER to obtain triangle i.d. of adjacent triangle.

m. MAPPER Draws material map of system.

n. CONVT Converts m-digit integer to character representation, used in MAPPER.

(2,0) GRIND2 Calls overlay (2,1), (2,2) and (2,3). Controls outer iteration process.

1. REBAL Calculates rebalance factors for both inner and outer iterations, for either fine mesh or whole system. Called by INNER and TESTS.

(2,1) 2. GRIND21 Calls INITAL, INITQ, and INITF.

a. INITAL Performs adjoint reversals of fission spectrum and velocities, mixes cross sections, initializes fission spectrum, initializes source by call to INITQ, generates total cross-section array, calculates fission source by call to INITF.

b. INITQ Generates source rebalance information, normalizes sources and related quantities if necessary, prints sources if desired.

c. INITF Calculates and normalizes fission array.

(2,2) 3. GRIND22 Calls OUTER.

a. OUTER Calculates source to group and rebalance source, calls INNER, calculates new fission rate.

b. INNER Calculates total source to group, sweeps over angles, performs sweep of spatial mesh by call to SWEEP, checks for convergence of inner iterations, applies rebalance factors calculated by REBAL, accumulates outer iteration rebalance arrays.

c. NEWAB Accumulates effective absorption array for outer rebalance.

d. SWEEP Performs sweep over the spatial mesh a band at a time, performs necessary rotations to standard right triangle, solves for angular flux on each triangle, stores outgoing flux for reflecting and special boundary conditions by call to STORBC, stores angular flux on last inner iteration.

e. SETBC Multiple entry-point routine to treat boundary conditions on the band sweeps. Entry SETBCB sets boundary condition on bottom of band, entry SETBCT sets boundary condition on top of band, entry SETBCE sets boundary condition on ends of band, and entry STORBC stores angular flux in appropriate array at end of sweep on a band.

(2,3) 4. GRIND23 Calls TESTS.

a. TESTS Calculates rebalance factors for outer iteration acceleration, makes k_{eff} reductions, computes neutron balance, checks for outer convergence, applies rebalance factors and calls NEWPAR for eigenvalue searches.

b. NEWPAR Computes new parameters for implicit eigenvalue search.

(3,0)	OUTPUT3	Controls prints, interface file output, and edits. Takes final dump.
(3,1)	1. OUTPT31	Calls FINAL.
	a. FINAL	Prints scalar flux and moments and fission rate.
(3,2)	2. OUTPT32	Calls EDCALL.
	a. EDCALL	Generates net leakage by call to GENFLO, performs zone edit by call to ZEDIT, performs point edit by call to PEDIT.
	b. GENFLO	Generates net leakages from each triangle by group.
	c. ZEDIT	Calculates zone balance tables, macroscopic activities, and power densities, calculates constituent and microscopic activities if desired.
	d. BANDED	Generates IBAND array again for use in EDMAP.
	e. EDMAP	Draws zone edit map of system.
	f. CONTV	Converts m-digit integer to character representation, used in EDMAP.
	g. PEDIT	Prints point scalar flux and components from angular flux, computes microscopic activities if desired.
	h. ARRMAP	Draws map of point arrangement on a triangle.
	i. CONVRT	Converts 2-digit integer to character representation. Used in ARRMAP.
(3,3)	3. IFOUT	Creates interface output files.
	a. IFRITE	Writes or prints interface file data.

TABLE V

RELATION OF PROBLEM VARIABLES TO PROGRAM MNEMONICS

	Program Mnemonic	Problem Variable	Refer to
1.	P(N,M)	$R_n^k(\mu_m,\phi_m)$	Eqs. (4), (14)
2.	C(Row,Nuclide)	σ_a, $\nu\sigma_f$, σ_t, $\sigma_{s,nh\to g}$	Section III.B.1
3.	WGT(M)	w_m	Section II.B.3
4.	COSMU(M)	μ_m	Section II.B.3
5.	COSETA(M)	η_m	Section II.B.3
6.	CTOT(I)	$\sigma_{t,i}$	Eq. (11)
7.	F(I)	f_i	Eq. (35)
8.	AF(K,I)	$\psi_i^{(k)}$	Eq. (20)
9.	S(N,K,I)	Source to a group	Section II.D.4
10.	FLUX(N,K,I)	Flux moments	Section II.D.4
11.	FISSA(K,I)	Fission source to a group	Section II.D.4

TABLE VI

CONTENTS OF BLANK COMMON BLOCK IA

Position	Name	Pointer for Array	Remarks
1	ITH		Theory
2	ISCT		Scattering order
3	ISN		Source of S_n constants indicator
4	IGM		Number of groups
5	NP		Finite element polynomial order
6	JT		Number of triangle bands
7	IBL		Left boundary specification
8	IBR		Right boundary specification
9	IBB		Bottom boundary specification
10	IBT		Top boundary specification
11	IEVT		Eigenvalue type specification
12	ISTART		Flux input option
13	MT		Total number of materials
14	MIN		Total number of input nuclides from both library and cards
15	MS		Number of mixture instructions
16	IHT		Position in table of total cross section
17	IHS		Position in table of self-scatter cross section
18	IHM		Cross-section table length
19	IQOPT		Source input options
20	IQAN		Distributed source anisotropy order
21	ISDS		Continuous/discontinuous model indicator
22	IACC		Rebalance option indicator
23	IPVT		k_{eff} or alpha parametric eigenvalue indicator
24	ISDB		System/fine-mesh buckling indicator
25			Not used
26	IITL		Maximum number of inner iterations
27			Not used
28			Not used
29			Not used
30			Not used
31	IEDOPT		Edit option indicator
32			Not used
33			Not used
34			Not used
35			Not used
36			Not used
37	EV		Eigenvalue guess
38	EVM		Eigenvalue modifier
39	PV		Parametric value of k_{eff} or alpha
40	XLAL		Search lambda lower limit
41	XLAH		Search lambda upper limit
42	XLAX		Fine-mesh search precision

43	EPS(EPSO)		Convergence precision and outer convergence precision
44	EPSI		Inner convergence precision = EPSO
45	EPSR		Within-group rebalance convergence precision = 10^{-4}
46	EPSX		Whole-system rebalance convergence precision = EPSO
47			Not used
48	NORM		Normalization amplitude
49	POD		New parameter modifier
50			Not used
51	IUP		IHS-IHT-1 (upscatter indicator)
52	IHF		IHT-1 (position of $\nu\sigma_f$ in cross-section table)
53	IHA		IHT-2 (position of σ_a in cross-section table)
54	IHTR		IHT-3 (position of σ_{tr} transport cross section in cross-section table, if present)
55	IHNN		IHT-4 (position of $\sigma_{n,2n}$ cross section in cross-section table, if present)
56	NPPT		Number of points per triangle actually stored
57	MM		Number of quadrature directions
58	NM		((ISCT+1)*(ISCT+2))/2, number of anisotropic components of flux
59	NMQ		((IQAN+1)*(IQAN+2))/2, number of anisotropic source components
60	NPP		Sum NP+1
61	NPT		(NP+1)(NP+2)/2, number of points per triangle
62	IGP		Sum IGM+1
63	NTC		Total number of triangles in system
64	ITMAX		Maximum number of triangles on one band
65	NTX		Total number of x-coordinates
66	NTF		Total number of fluxes stored, NPPT*NTC
67	NTQ		Total number of sources stored, NPPT*NTC
68	NUPB		Number of upward pointing triangles on bottom band
69	NDNT		Number of downward pointing triangles on top band
70	NMM		Product NM*MM
71			Not used
72	IHMT		Product IHM*MT
73			Not used
74	ISCP		Sum ISCT+1
75	LNBT		Length of BT array
76	LNBB		Length of BB array
77	LNBL		Length of BL array
78	LNBR		Length of BR array
79	ICLIM		Length of the C and AAJ blocks
80	LIT	IT(JT)	Number of triangles on each band
81	LITT	ITT(JT)	Indicator for orientation of first triangle on each band
82	LC	C(IHM,MT)	Cross sections for a group
83	LKC	KC(JT+1)	Pointer for first triangle i.d. on each band
84	LKX	KX(JT+2)	Pointer for first x-coordinate on each node line
85	LIB	IBAND(NTC)	Pointer to adjacent triangle on band above or below
86	LBND	BOUND(NPP*ITMAX/2)	Angular flux array on band boundary
87	LIAF	IAF(NPT,6)	Angular flux index array

88	LQ	Q(NM,NPT,NTC)	Distributed source
89	LX	X(NTX)	X-coordinates of triangle vertices
90	LAE	AE(ITMAX+1)	Area elements for a single band
91	LAS	AS(ITMAX+1)	Sign of AE times 1.0
92	LAF	AF(NPT,ITMAX+2)	Angular flux on a single band
93	LBT	BT(NPP,NDNT,MM)	Top boundary angular flux array, actual length is LNBT
94	LBB	BB(NPP,NUPB,MM)	Bottom boundary angular flux array, actual length is LNBB
95	LBL	BL(NPP,JT,MM)	Left boundary angular flux array, actual length is LNBL
96	LBR	BR(NPP,JT,MM)	Right boundary angular flux array, actual length is LNBR
97	LP1	P1(NPT,NPT)	Inner product array, (Lagrange, Type 1 weight) polynomials
98	LB2	P2(NPT,NPT)	Inner product array, (Lagrange, Type 2 weight) polynomials
99	LPX1	PX1(NPT,NPT)	Inner product array, (x-derivative of Lagrange, Type 1 weight) polynomials
100	LPX2	PX2(NPT,NPT)	Inner product array, (x-derivative of Lagrange, Type 2 weight) polynomials
101	LPY1	PY1(NPT,NPT)	Inner product array, (y-derivative of Lagrange, Type 1 weight) polynomials
102	LPY2	PY2(NPT,NPT)	Inner product array, (y-derivative of Lagrange, Type 2 weight) polynomials
103	LB1	B1(NPP)	Triangle boundary flux
104	LB2	B2(NPP)	Triangle boundary flux
105	LFL	FLUX(NM,NPT,NTC)	Flux components
106	LFLA	FLUXA(NTC)	Scalar flux from previous inner iteration
107			Not used
108	LFIS	FISS(NTC*NPT)	Same origin as FISSA
109	LFISA	FISSA(NTC*NPT)	Fission source
110			Not used
111	LDC	IDCS(NTC)	Cross-section identification number for each triangle
112	LCC	CC(NPT,NPT)	Matrix inverted for angular fluxes on each triangle
113	LB	B(NPT)	Source vector for angular fluxes on each triangle
114	LPRT	PRT(NPP,NPT)	Matrix for contribution to source vector B from orientation 1 triangle right boundary
115	LPBT	PBT(NPP,NPT)	Matrix for contribution to source vector B from orientation 2 triangle bottom boundary
116	LPLT	PLT(NPP,NPT)	Matrix for contribution to source vector B from orientation 2 triangle left boundary
117	LPI	PI(NPT)	Integrals of Lagrange polynomials
118	LHY	HY(JT)	Band heights
119	LW	WGT(MM)	Direction weights
120	LCM	COSMU(MM)	X-direction cosines
121	LCE	COSETA(MM)	Y-direction cosines
122			Not used
123			Not used
124	LP	P(NM,MM)	Spherical harmonic functions
125	LPBI	PBI(NPP)	Integrals of one-dimensional Lagrange polynomials
126	LPF	PF(NPT)	Equals PI(NPT) or 1.0 if ISDS = 1 or 0.
127			Not used

128	LMN	MIXNUM(MS)	Input mixture numbers (conditional on MS.GT.0)
129	LMC	MIXCOM(MS)	Input mixture instructions (conditional on MS.GT.0)
130	LMD	MIXDEN(MS)	Input mixture densitites (conditional on MS.GT.0)
131	LF	F(NTC)	Fine-mesh rebalance factors
132			Not used
133			Not used
134			Not used
135	L7LO	FLO(3,NTC)	Fine-mesh inward flows
136	LAB	AB(NTC)	Fine-mesh absorption removal rate
137	LQQ	QQ(NTC)	Fine-mesh source
138			Not used
139			Not used
140			Not used
141			Not used
142			Not used
143			Not used
144	LHA	HA(1TMAX)	Used in subroutine REBAL for inversion
145	LGA	GA(ITMAX)	Used in subroutine REBAL for inversion
146	LQG	QG(IGP)	Space integral of Q
147	LQGB	QGB(IGP)	Space integral of boundary source
148			Not used
149			Not used
150			Not used
151			Not used
152			Not used
153			Not used
154			Not used
155			Not used
156			Not used
157	LBAL	BAL(IGP)	Group neutron balance
158	LCHI	CHI(IGP)	Input fission spectrum
159	LCHIA	CHIA(IGP)	Fission spectrum used in the calculation
160	LVEL	VEL(IGP)	Group velocities
161			Not used
162			Not used
163			Not used
164			Not used
165	LSOU	SOURCE(NM,NPT,NTC)	Total souce in a group (same origin as Q)
166			Not used
167	LAA	AAJ(MT)	Effective absorption cross section
168	LBHT	BHT(NTC)	Space dependent buckling height
169			Not used
170			Not used
171			Not used
172			Not used
173			Not used

174	LCTOT	CTOT(NTC)	Effective total cross section
175	JCONV		Final convergence indicator
176	TN2N		N,2N reaction term used in balance equations
177	XLAPP		Value of lambda from sequence of outer iterations previous to that of XLAP
178	XLAP		Value of lambda from previous sequence of outer iterations
179	ICNT		Iteration trigger used in NEWPAR
180	E2		Temporary storage
181	E1		Temporary storage
182	EVPP		Eigenvalue from cycle of outer iteration previous to that of EVP
183	EVP		Eigenvalue from previous sequence of outer iterations
184	E4		Temporary storage
185	NGO		Return indicator set in NEWPAR
186	ALAR		Value of lambda from previous outer iteratiou
187	FT		Total space-integrated fission source in a group
188	IITNO		Inner iteration number
189	TS		Total integrated source to a group
190	G		Number of current group (integer)
191	ICONV		Secondary convergence indicator
192	NGOTO		Return indicator set in TESTS
193	E3		Temporary storage
194	EVS		Slope used in eigenvalue search
195	IITOT		Total number of inner iterations
196	ALA		Parameter lambda
197	TIN		Time
198	FTP		Previous fission total
199	IFN		Fission calculation indicator set in INITAL
200	OITNO		Outer iteration number
201	NPQ1		Number of angles in quadrant 1
202	NPQ2		Number of angles in quadrant 2
203	NPQ3		Number of angles in quadrant 3
204	NPQ4		Number of angles in quadrant 4
205	LMBT		First dimension of BT array
206	LMBB		First dimension of BB array
207	LMBL		First dimension of BL array
208	LMBR		First dimension of BR array
209			Not used
210			Not used
211			Not used
212			Not used
213			Not used
214			Not used
215	NTAF		Computed by subroutine KSET, not used in code
216	ERR		Scalar flux error from comparison with previous flux
217			Not used
218			Not used
219			Not used

220		Not used
221		Not used
222		Not used
223		Not used
224		Not used
225	NLIMIT	Not used
226	IFLAG	Not used
227 through 246		Not used
247	TIMACC	Accumulated problem running time
248	MCRRDS	Signed number of input nuclides requested from the code dependent input file
249	NOSGUP	Sigma up included in cross-sections indicator
250	IOLYCS	Overlay indicator (first digit gives primary overlay and second digit gives secondary overlay when read in octal)

TABLE VII

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.

Position	Name	Contents and Remarks
1	IDUSE	A vector used for the title of the problem in A4 format (length of 18 words)
2	LAST	Length of common block A used by the current problem
3	LASTEC	Length of LCM used by the current problem
4	IGCDMP	Group number of restart dump
5	IPSO	LCM pointer of the source-to-the-group block which is calculated by subroutine OUTER and used by subroutine INNER
6	LTSO	Length of the source-to-the-group block
7	IPFX	LCM pointer for the first group of the flux block (each group block contains the three-dimensional flux array as well as boundary arrays stored consecutively)
8	LTFX	Length of the flux block for a group
9	LXFX	Length of the three-dimensional flux array
10	IPXS	LCM pointer for the first group of the cross-section block (each group block contains the cross-section array, the effective absorption vector and the spatial total cross-section array)
11	IPXSCT	LCM pointer for the first group of the spatial total cross-section array within the cross-section block
12	LTXS	Length of the cross-section block for a group
13	LTOXS	Length of the cross-section array for a group
14	LTAXS	Length of the cross-section array and the effective absorption vector for a group
15	IPQS	LCM pointer for the first group of the Q-source block
16	LTQS	Length of the Q-source block for a group (zero when not a Q-calculation)
17	IEREC	Not used
18	I2	Final flux print indicator (0/1/2 = all/isotropic/none)
19	I4	Final fission print indicator (0/1 = yes/no)
20	I6	Not used
21	IFO	Interface output indicator (0/1 = no/yes)
22	IPBS	LCM pointer for the first group of the boundary source block
23	IPAB	LCM pointer for the absorption block used in the outer rebalance
24	LTAB	Length of the absorption block used in the outer rebalance
25	IPFLO	LCM pointer for the inward flows block used in the outer rebalance
26	LTFLO	Length of the inward flows block used in the outer rebalance
27	IPQQ	LCM pointer for the group summed Q-source block used in the outer rebalance
28	LTQQ	Length of the group summed Q-source block used in the outer rebalance

TABLE VIII

CONTENTS OF NAMED COMMON BLOCK FWBGN2

The named common block FWBGN2 consists primarily of those indicators used by the program but not vital to restart. Parameters which define options are set in program TRIPLET.

Position	Name	Contents and Remarks
1	TIMBDP	Minimum time between periodic dumps (0/F = no periodic dumps/seconds)
2	TIMSLD	Elapsed time since last dump
3	TIMOFF	Floating-point form of the input fixed-point parameter ITLIM
4	MAXLEN	Maximum length of the main common data block A
5	MAXECS	Maximum length of LCM storage available to the problem
6	LENMCB	Length of the named common block FWBGN1 which must be saved for restarts
7	LENCIA	Length of the common parameter block IA which must be saved for restarts
8	IFNOVY	Overlay file name for CDC machine usage (given in left justified, zero fill Hollerith form)
9	IRCOVY	Recall overlay indicator for CDC machine usage (0/6HRECALL = no reloading of overlay when in core/reloading overlay when in core)
10	I1	Full flux guess input print indicator (indicator effective when ISTART is 2 or -2, 0/1 = yes/no)
11	I3	Cross-section print indicator (0/1/2 = all/mixed/none)
12	I5	Input source print indicator (0/1/2/3 = all/unnormalized/normalized/none)

TABLE IX

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.

Position	Name	Contents and Remarks
1	NERROR	Parameter input-error indicator
2	ITLIM	Fixed-point time problem removal value (0/N = no/seconds)
3	MCR	Number of nuclides requested from the code dependent input file (minus if FIDO format)
4	MTP	Number of library nuclides requested from ISOTXS interface file (MTPS*(ISCT+1))
5	MTPS	Number of nuclide sets requested from ISOTXS interface file (each set yields ISCT+1 blocks)
6	NISOXS	Number of isotopes in the set as read from the ISOTXS file
7	LMTP	Core pointer for the ISOTXS interface file position number data block
8	NEXTER	Fetch next case indicator (if any error was detected after all input was successfully read, the next problem is fetched)
9	LIMIT	Maximum storage length required by input cross sections and calculation of anisotropic scattering coefficients
10	LENCLR	Length of the partial block to be cleared during input

TABLE X

CONTENTS OF NAMED COMMON BLOCK CSWEEP

The named common block CSWEEP is used to pass the current value of variables from subroutine INNER to subroutine SWEEP and to subroutine SETBC without using a calling sequence entry.

Position	Name	Contents and Remarks
1	U	Direction cosine u_m during angle sweep
2	E	Direction cosine η_m during angle sweep
3	WGT	Direction weight w_m during angle sweep. Variable name is WATE in subroutine INNER
4	M	Do-loop index of angle sweep
5	I	Triangle index on each band of spatial mesh sweep, I = 1, 2, ..., IT(J)
6	J	Band index of spatial mesh sweep, J = 1, 2, ..., JT
7	ID	Triangle identification number of spatial mesh sweep, ID = KC(J) + I - 1

TABLE XI

CONTENTS OF NAMED COMMON BLOCK UNITS

The named common block UNITS contains the symbolic names of all input, output, and scratch devices required by TRIPLET which are set in the main program TRIPLET.

Position	Name	Contents and Remarks
1	NINP	Problem code dependent decimal input
2	NOUT	Problem decimal output
3	NAFLUX	Angular flux by group
4	NDUMP1	First restart dump unit
5	NDUMP2	Second restart dump unit
6	NEXTRA	Scratch unit
7	NEDIT	Edit input storage
8	IAFLUX	Interface form of angular flux
9	ITFLUX	Interface form of total flux
10	ISNCON	Interface form of S_n constants
11	IFIXSR	Interface form of inhomogeneous source (Q-source)
12	ISOTXS	Interface form of multigroup cross-section file ISOTXS
13	NLEAK	Triangle net leakages by group

5. Machine Dependent Subprogram

a. LCM System Routines

LCM (large core memory) is a large bulk memory from which blocks of words may be quickly transferred to or from SCM (small core memory). This random bulk memory is accessed through two system routines - ECRD (transfers LCM to SCM) and ECWR (transfers SCM to LCM) - which process consecutive words of SCM and consecutive words of LCM given an SCM address and a pointer value for LCM. The pointer value given may be thought of as the index of a container array. To read from or write into a block of core, it is necessary to provide the read/write routines with the core origin, the LCM pointer value and the number of consecutive words to be transferred. For example, if we consider reading the entire FLUX block for group IG from LCM to SCM, we would have the FORTRAN IV statements

CALL REED(0,IPFX+(IG-1)*LTFX,FLUX,LTFX,1) and

CALL ECRD(FLUX,IPFX+(IG-1)*LTFX,LTFX,IER) .

In these statements FLUX is the SCM container array, IPFX+(IG-1)*LTFX is the location of the first word of the IGth group flux array in LCM, and LTFX words are transferred. IER is an error indicator. On the CDC 6600 Extended Core Storage (ECS) plays the role of LCM. On the IBM-360/195 ECRD and ECWR are replaced with core-to-core transfers.

b. General System Routines

Additional system routines required by the code are SECOND (obtains current time), DATE1 (obtains current date), ATAN (arctangent), SQRT (floating-point square root), EXIT (returns control to system for next job), and COS (cosine).

Use of an end of file test is made in INPUT11 to detect the last case of a sequence of cases. The test must be replaced by an equivalent statement to obtain a normal exit.

B. External and Internal Data Files

All files used for input, output and scratch data are referred to by symbolic name throughout the code. The user may easily change the physical unit assigned a file by modification of the symbolic name which is initialized in the main program of TRIPLET. Table XII indicates the files required by TRIPLET.

TABLE XII

TRIPLET FILE REQUIREMENTS

Name	Logical Unit (CDC Machines Only)	Contents	Remarks
NINP	10	Problem code dependent decimal input	The user may wish to equate this file to the system input file.
NOUT	9	Problem decimal output	The user should equate this file to the system decimal printed output file.
NAFLUX	6	Binary angular flux by group generated only on a special last outer iteration	The contents for each group and direction consists of JT records, each of length IT(J)*NPT, written a band at a time.
NDUMP1	7	Restart dump	This unit is used to receive the first restart dump when the problem is not restarted from a previous dump. The unit must contain the restart dump information when the problem is restarted and will then be used to receive the second restart dump (NDUMP2 receives the first dump).
NDUMP2	5	Restart dump	Second restart dump unit.
NEXTRA	18	Scratch file	The file is used in the decimal mode for Hollerith conversions rather than the core-to-core conversions given by the FORTRAN statements of ENCODE and DECODE of CDC machines.
NEDIT	17	Edit input storage	To save core, edit input is stored on this file until time of edit.
IAFLUX	31	Interface form of angular flux (either adjoint or regular)	Output of the angular fluxes in interface form is placed on this file. The file is rewound prior to processing the fluxes and an end of file is placed on the file after the last write. Data for one problem only is kept on this file.

ITFLUX	30	Interface form of total flux (either adjoint or regular)	The code requires that this unit be used when a flux guess is requested from the total flux interface file. The unit is rewound and the records of the first file are used as the input guess. The interface form of the total flux is prepared on this file as problem output by rewinding the file and writing the file in standard format. An end of file is placed on the file after the last write instruction.
ISNCON	32	Interface form of S_n constants	When the file is used as input, the file is rewound and read. When used as output the file is rewound and written, including an end of file.
IFIXSR	33	Interface form of source	This file is used only as input for the cell centered inhomogeneous source. Boundary sources (if any) are obtained from the code dependent input file.
ISOTXS	34	Interface form of the cross-section multigroup file ISOTXS	This file is only used as input when cross sections are requested from an interface file library.
NLEAK	8	Fine-mesh net leakages	This file is used to transfer the net leakages for each triangle from GENFLO to ZEDIT. The file contains IGM records of length NTC words.
N5	40	Scratch storage for FIDO subroutine which appears only in the IBM code version	Necessary for reading FIDO format cross sections on IBM-360 machines only.

C. Hardware Requirements

The TRIPLET code does not require any special hardware. The LASL CDC 7600 provides 65K (decimal) SCM and 512K LCM 60-bit words. Only 370K LCM are available to the user with the operating system and buffers using the remainder. Type 7638 disk units provide 84 million decimal words of peripheral store per unit.

D. Software Requirements

1. CDC Machines

The code was designed to operate on the CDC 7600 under the CROS operating system[8] which was developed at Los Alamos. The system uses the CDC RUN compiler with a CDC optimizer attached.

The code operates on the CDC 6600 under SCOPE 3.1 which has been heavily modified[9] at Los Alamos, and the CDC RUN compiler is again used.

On both machines, the disk units provide storage for input, output, scratch, and resident files.

2. TRIPLET for the IBM-360

TRIPLET was prepared for the IBM-360/195 from the CDC 7600 version of the code. The IBM-360 version operates in the four byte floating-point mode. The CDC 7600 version of the code was prepared so that conversion to the IBM-360 would involve as few changes as possible and so that an eight byte floating-point version of the code could easily be made, if needed. Entries of named common blocks were ordered to make this possible, and formats for reading input title data were changed to A4 formats.

The major change made in the conversion of the code was in the treatment of peripheral storage. The vast amount of fast core available on the IBM-360 is one of the cheaper resources of that machine. It was thus decided to place the data normally kept in LCM (large core memory) in an expanded container array. This was accomplished by using part of the container array (indexed from 1 through LAST) as small core (SCM) and the remainder as LCM (indexed starting at LAST+1 and ending at MAXLEN). The system routines ECRD and ECWR of the CDC 7600 were replaced by simple routines which move data to and from sections of the container array. It was thus possible to keep the LCM pointer structure of the code with no change in logic and with little cost in time for data movement. It should be noted that the amount of core neeed (LAST) is computed first and the amount of LCM needed is computed as MAXLEN minus LAST. The size of the container array is changed simply by changing the dimension of common block A and assigning a new value to MAXLEN. Both changes need be made only in program TRIPLET.

In the edit and interface output sections of the code, the amount of SCM required is recomputed and tested against LAST. In rare cases these tests may fail, but enough core may exist in the container array for both SCM and LCM. Then, the value LAST may be reset and that portion of the container array corresponding to LCM moved to accommodate the increased requirement for SCM.

Because of the amount of core storage available on the IBM machines, it was found to be computationally advantageous to combine the computational overlay GRIND2 with its secondary overlays.

In addition to storage reorganization, the following changes were made to effect the IBM conversion of TRIPLET:

1. Dummy subroutines were substituted for system subroutines DATE1 (returns date as an A8 word) and SECOND (returns floating-point value of the current time). The dummy routine for SECOND must be replaced by a local system routine or the periodic and time limit dump options will not work.
2. A subroutine FIDO was added to process the FIDO cross-section format. The CDC 7600 algorithm to read this format uses a rewind command. On the IBM system a prohibitively large amount of wait time was required because of the rewind command.
3. Hollerith 6H constants throughout the code were typed as real eight byte data, including subroutine call list variables. Subroutine WRITE uses one such variable as either an integer (four bytes) or a 6H constant (eight bytes), and this situation was treated by equivalencing statements.
4. The IF(EOF) CDC job termination test was replaced with a read with the IBM END parameter.
5. CDC overlay cards were removed, overlay programs were changed to subroutines, calls to overlay segments were replaced with calls to the subroutines, and the setting of overlay parameters IFNOVY and IRCOVY was eliminated.
6. Several options were added to subroutine REED to treat interface file data. All such data is assumed to be four-byte words, but the specification record of such files is three eight-byte words plus one four-byte integer. Two arrays of the ISOTXS file contain mixed 6H data (8 bytes), and floating point and integer (4 bytes). Finally, reading the ISNCON file required skipping a portion of the record before reading additional data. All of these problems were treated by adding options to REED.
7. Shortened names (six character maximum) were required for the following routines:

CDC	IBM-360
TRIPLET	(Main program)
INPUT11	INPT11
INPUT12	INPT12
INPUT13	INPT13
INPUT14	INPT14
INPUT15	INPT15
GRIND21	GRID21
GRIND22	GRID22
GRIND23	GRID23
OUTPUT3	OUTPT3
OUTPT31	OUTT31
OUTPT32	OUTT32

E. Programming Considerations

1. Storage Management

a. Variable Dimensioning

A single container array, A, in common is used for the blocks of data required in executing a problem. The storage of all data is consecutive and compact in the A array so that the size of a problem is limited by the total storage required rather than by the size of individual parameters. A pointer word is associated with each data block and is used to index A to locate the block. For example LFL is the first word address of the flux block in A and A(LFL) is the first word of the flux array. When subroutine calls are written, the address of a data block, say A(LFL), is passed through the argument call. In the subroutine the data block is variably dimensioned so that it may be easily indexed by its subscripts, e.g. FLUX(N,I,J).

b. Allocation of Large Core Memory (LCM)

The allocation of storage in large core memory (LCM) is handled in the same manner as core storage. Most of the group-dependent arrays are stored in LCM so the dimensionality is IGM times the core requirement of the array. For example, there are IGM*NM*NPPT*NTC LCM locations required for FLUX(NM,NPPT,NTC).

Certain blocks of data are stored contiguously in core so that they may be read in and out of LCM in a single stream. For example, the flux block includes FLUX(NM,NPPT,NTC), BT(NPP,NDNT,MM/2), BB(NPP,NUPB,MM/2), BL(NPP,JT,MM/2), and BR(NPP,JT,MM/2) when the boundary conditions are other than vacuum. The first word of this block is LFL, and the last word is LFLA-1. The cross-section block includes the cross sections C(IHM,MT), the effective absorption AAJ(MT) and the total cross section CTOT(NTC). The first word of this block is LC, and the last word is LQ-1. A complete list of LCM storage is given in Table XIII.

c. Computation of Required Storage

The easiest way to compute the storage required by a problem is to load the problem for a short run and let the code compute LAST, the amount of SCM and LASTEC, the amount of LCM. The computation is made very early in problem execution and this result is printed before most of the data is read. An approximate formula for LAST is

LAST = MT*IHM+(NPT+2)*$\max\limits_{j}$ IT(j)+NTX+5*JT
+3*NTC*(NPT+1) + Conditional Blocks
+ Boundary Array Blocks

where the variables in the formula are defined in Table VI. The sizes of the conditional blocks are:

NTC	for ISDB=1
7*NTC	for IACC=1

The sizes of the boundary array blocks are:

NDNT*NPP*MM/2	for IBT=1,
NDNT*NPP*MM	for IBT=3,
NDNT*MM	for IBT=-3,
NUPB*NPP*MM/2	for IBB=1,
NUPB*NPP*M.	for IBB=3,
NUPB*MM	for IBB=-3,
JT*NPP*MM/2	for IBR=1, IBR=2 or IBL=1,
JT*NPP*MM	for IBR=3 or IBL=3,
JT*MM	for IBR=-3 or IBL=-3.

TABLE XIII

LCM STORAGE PARAMETERS

LCM First Word Address	Length Per Block	No. of Blocks	Contents
IPXS	LTXS = NTC+MT*IHM+MT	IGM	Cross-section blocks by group
IPFX	NM*NPT*NTC	IGM	Scalar flux and moments
IPBS	LTFX[a] = NM*NPT*NTC	IGM	Boundary flux or boundary source
IPSO	LTSO = NM*NPT*NTC	1	Source to group including within group scatter
IPQS	LTQS = NM*NPT*NTC	IGM	Q-source blocks by group, present only if IEVT=0
IPAB	LTAB = (NTC or 1)	1	Absorptions plus net leakages for (fine-mesh or system) rebalance
IPFLO	LTFLO = 3*(NTC or 1)	1	Inward flows for (fine-mesh or system) rebalance
IPQQ	LTQQ = (NTC or 1)	1	Total inhomogeneous source for (fine-mesh or system) rebalance

[a] LTFX is the length of the flux arrays (NM*NPT*NTC) plus the length of the boundary flux arrays, which depends on the boundary conditions. The lengths of the boundary arrays are given in Sec. IV.E.1.c. for the various boundary condition options.

The amount of LCM is given by

LASTEC = IGM(MT*(IHM+1)+NTC*(1+2*NM*NPT)
+ Boundary Array Blocks)
+ 2*NM*NPT*NTC + Conditional Block

where the conditional block size is

5*NTC or 5	for IACC=1 or 0

The above formulas are all based on use of the discontinuous model. The storage for the continuous model may be estimated from these formulas by setting NPT=1.

d. Temporary Storage Requirements

The amount of storage actually calculated for LAST is the maximum of three quantities. The first of these is the total amount of SCM required for problem execution and the other two are temporary storage requirements, one for input of cross sections and the other for calculation of anisotropic scattering coefficients. Usually, the problem data requirement is much larger than the temporary storage requirement, but occasionally the input cross-section requirement (IGM*IHM*MIN words) is largest.

A test is made for each of the temporary storages and the user is informed if more SCM is required than is available. Additional tests are made prior to the processing of each zone or point edit in EDCALL and prior to the processing of the interface output in IFOUT.

e. Overstorage of Data in Core

In TRIPLET, a certain amount of overstorage is used to reduce the total amount of small core memory (SCM) required; i.e., more than one array may reside in the same SCM locations as the problem progresses. The most important example of this is that the large arrays Q(NM,NPPT,NTC) and SOURCE(NM, NPPT,NTC) are stored in the same spaces. The Q array is used in OUTER to generate the total source to a group, including the inhomogeneous source, if any. This source is then written in LCM. In INNER this source is read into SCM after each inner iteration to initialize the SOURCE array. Then the rest of the group source is added to SOURCE.

When input cross sections are read, they are read into location LQ, and when anisotropic scattering spherical harmonic coefficients are calculated, storage beginning at location LQ is used. In each of these cases, the amount of storage required is checked (previous section), and the core is reinitialized after the operation is complete. Similar overstorage is performed for temporary arrays used by the POLY subroutine and is performed when the material map is written (subroutine MAPPER) for a problem not using fine-mesh rebalance.

2. Restart Tape Composition

The restart dump is composed of the following records: restart parameter information, named common block FWBGH1, common block IA, data common block A (only the portion used by the problem), and angular flux records. The angular flux information is only generated in the last pass of the calculation for groups one through the current group. The number of data entries is the first word of each record on the dump tape. The final dump contains the current group (IGCDMP) value of zero.

The restart parameter information is a vector which contains five words in the following order:

1.	IGM	Number of groups.
2.	NORDM	Number of LCM records.
3.	NORDAF	Number of angular flux records.
4.	NTC	Number of triangles.
5.	NM	Number of flux components.

3. Standard Interface Files

The standard interface files read and written by TRIPLET are version II files.[4] As far as possible the codes which process these files are subroutines. The user should note that in reading, the very first entry on a file is used as the input, and in writing, the file is rewound prior to the output of data. No physical unit distinction is made between regular and adjoint input or output files. If a standard interface file is used to provide a flux guess, and a standard interface file flux output is requested, the input file information is destroyed.

REFERENCES

1. K. D. Lathrop and F. W. Brinkley, "TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport," Los Alamos Scientific Laboratory report LA-4848-MS (July 1973).

2. Proposed American Nuclear Society Standard 10.3, "Guidelines for the Documentation of Digital Computer Programs," to be published.

3. G. I. Bell, G. E. Hansen, and H. A. Sandmeier, "Multitable Treatments of Anisotropic Scattering in S_n Multigroup Transport Calculations," Nucl. Sci. Eng. 28, 376 (1967).

4. B. M. Carmichael, D. A. Meneley, and D. R. Vondy, "Report of the Subcommittee on Standard Interface Files," prepared for the Committee on Computer Code Coordination and issued as Los Alamos Scientific Laboratory report LA-5324-MS.

5. W. W. Engle, Jr., "A Users Manual for ANISN, a One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering," Union Carbide Corporation report K-1693 (March, 1967).

6. K. D. Lathrop, "DTF-IV, A Fortran Program for Solving the Multigroup Transport Equation with Anisotropic Scattering," Los Alamos Scientific Laboratory report LA-3373 (November 1965). p. 42.

7. B. G. Carlson and K. D. Lathrop, "Transport Theory - Method of Discrete Ordinates," Chapter III of Computing Methods in Reactor Physics, Gordon and Breach, New York (1968), p. 257.

8. "Elementary Guide to the Control Data 7600," Ray Davenport, Ed., Programmer's Information Manual, Vol. 5A, Los Alamos Scientific Laboratory (1972).

9. Programmer's Information Manual, Vol. 4, CDC 6600, Anne Browning, Ed., Los Alamos Scientific Laboratory (1973).

10. W. H. Reed and T. R. Hill, "Triangular Mesh Methods for the Neutron Transport Equation," to be published in the Proceedings of the National Topical Meeting on Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, April 9-11, 1973, Ann Arbor, Mich.