LA-4769

c. 3

Triangular Mesh Difference Schemes for the Transport Equation

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

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

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 $3.00; Microfiche $0.95

LA-4769
UC-32
ISSUED: November 1971

Triangular Mesh Difference Schemes for the Transport Equation

by

Wm. H. Reed

TRIANGULAR MESH DIFFERENCE SCHEMES FOR THE TRANSPORT EQUATION

by

Wm. H. Reed

ABSTRACT

Present transport codes require that the physical system be described by an orthogonal mesh. This restriction leads to the simplest difference schemes but may require that an excessive number of mesh points be used to describe adequately a complicated system. Triangular mesh difference schemes for the transport equation are discussed in this report. These schemes preserve the simplicity of difference schemes on an orthogonal mesh yet permit a much finer representation of complicated geometries, for a given number of mesh cells. Solution of the triangular mesh difference equations is discussed, and the truncation error of these equations is derived.

I. INTRODUCTION

Solutions of the neutron transport equation are obtained most often by the method of discrete ordinates,[1] often referred to as the S_n method. This method represents a direct discretization of the integrodifferential transport equation. A number of computer codes[2-4] have been developed using the method in one and two space dimensions and in rectangular and curved geometries. All of these codes have utilized an orthogonal mesh, which we define as a mesh whose grid lines meet at right angles. The use of such an orthogonal mesh leads to the simplest difference equations but may require an excessive number of mesh points to describe complicated geometries adequately.

Consider the description of a circular region in x-y geometry as illustrated in Fig. 1. This is perhaps the simplest of complicated geometries. We attempt to describe Fig. 1 by using a standard orthogonal grid where all grid lines run parallel to the coordinate axes. Restricting to fewer than 50 mesh cells, we obtain the representation of Fig. 2, which is poor, but is the best that can be done under the above restriction. To obtain an adequate representation of the circle, on the order of 1000 mesh cells are needed. Of course, most geometries of physical interest are more complicated than Fig. 1, and many thousands of mesh cells are needed.

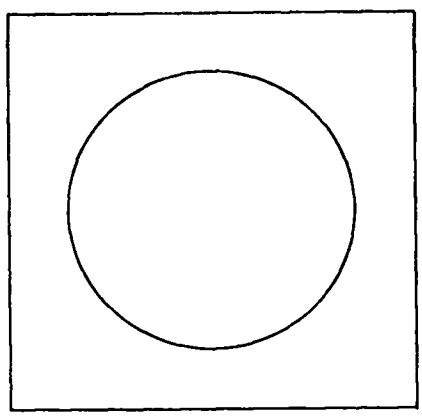


Fig. 1. A simple circular region in rectangular geometry.

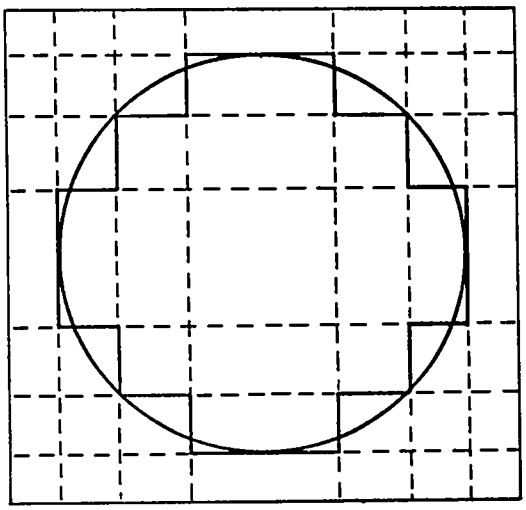


Fig. 2. An attempted description of a circular region with an orthogonal mesh of 49 mesh cells.

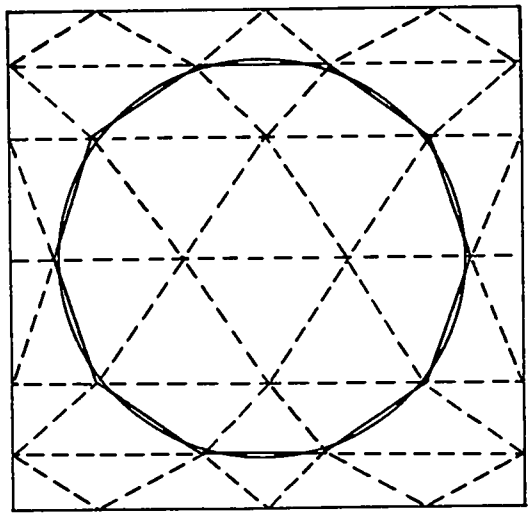


Fig. 3. An attempted description of a circular region with a triangular mesh of 46 cells.

The source of this difficulty lies in the requirement that mesh lines be parallel to coordinate axes, or, stated more properly, that the mesh be orthogonal. If this requirement is abandoned, then complicated geometries can be drawn easily. If, however, no regularity is preserved in the mesh grid, then the description of the mesh becomes a major problem and the difference equations become excessively complicated. A triangular mesh is a good compromise because it is flexible enough to represent the most complicated geometries and yet preserves the regularity vital to simplicity of the implementing code. We illustrate in Fig. 3 the flexibility of a triangular mesh by drawing with triangles the circular region shown in Fig. 1. It is clear that, with fewer mesh cells, triangles permit a much finer representation of such curved shapes than does an orthogonal grid.

To establish conclusively the above point, we present in Fig. 4 a triangular decomposition of a complicated geometry, which is shown in Fig. 5 with an attempted orthogonal representation.

We propose to develop a two dimensional discrete ordinates transport theory code based on a triangular mesh. We intend to incorporate into this code most features and options of the presently available two dimensional transport code TWOTRAN.[4] Some of these features are

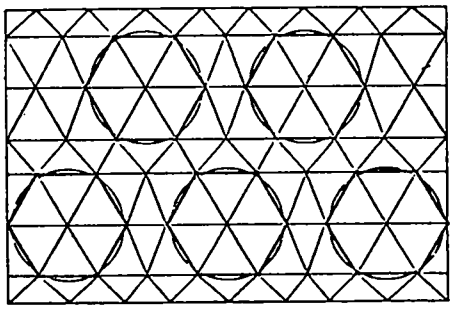


Fig. 4. Triangular representation of a complicated geometry.

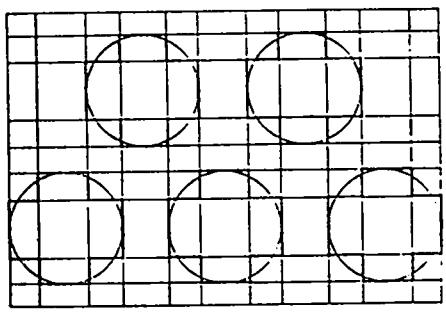


Fig. 5. Orthogonal representation of a complicated geometry.

1. (x,y) and (r,z) geometries,

2. Direct or adjoint calculations,

3. General order scattering anisotropy,

4. "Group at a time" solution so that storage requirements are independent of number of energy groups,

5. Flexible boundary conditions,

6. Inhomogeneous source or eigenvalue (k_{eff} or time) calculation,

7. Criticality searches on nuclide concentration or zone thicknesses,

8. Coarse mesh rebalancing of inner and outer iteration processes,

9. Input of cross sections from cards or disk file,

10. Built-in S_n constants,

11. Flexible restart procedures,

12. Flexible input for flux guesses and sources, and

13. Detailed editing capabilities.

In addition, we propose to include as an option a newly developed method designed to eliminate the ray effect. The ray effect is a severe spatial distortion of the neutron flux characteristic of the discrete ordinate method in two dimensional geometries. This effect is eliminated by adding a fictitious source to the discrete ordinate equations that forces these equations to yield solutions to the P_n equations.

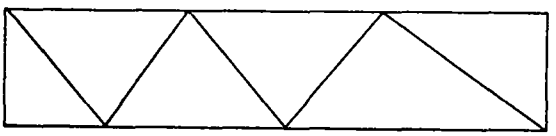
A neutron transport code featuring a triangular mesh and including the above options offers more flexibility to the reactor designer than do the present transport codes.

II. DIFFERENCE METHODS ON TRIANGULAR MESHES

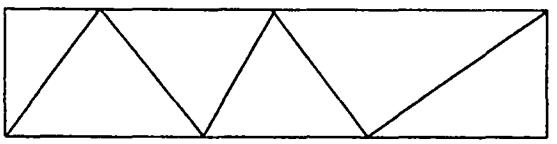
In this report, we consider only the "regular" triangular meshes, by which we mean that every interior node is the vertex of six adjacent triangles. No condition is imposed on boundary nodes. A further restriction on the mesh is the requirement that all nodes lie on horizontal lines extending through the system. The purpose of these restrictions is to simplify the description of the mesh without destroying its flexibility. All the examples of triangular meshes presented in the introduction are of the above type.

Let us assume, for the moment, that we are dealing with x-y rectangular geometries. To specify such a triangular mesh we must give the y coordinates of the horizontal lines and the x coordinates of the nodes along each line. The mesh is then completely determined by the direction of the first triangle on each band. Consider a simple example consisting of a single band with six nodes on each of the two horizontal lines forming the band. The two possible triangular arrangements are indicated in Fig. 6. We refer to a triangular mesh with the first triangle pointing upward as being Type 1. If the initial triangle points downward, the mesh is said to be Type 2. Thus, to completely determine a triangular mesh of the kind being considered, we must specify a type number for each band, in addition to specifying the coordinates of the nodes.

Having defined our mesh, we can now derive a set of finite difference equations. For simplicity, these equations will be derived only for the case of x-y geometry, although it is intended that the proposed code will also handle r-z cylindrical geometry. The analytic form of the transport equation in x-y geometry is given by



Type 1



Type 2

Fig. 6. The two possible arrangements of triangles on a single band.

$$\mu \frac{\partial \psi}{\partial x} + \eta \frac{\partial \psi}{\partial y} + \sigma \psi = S, \qquad (1)$$

where ψ is the neutron flux and μ and η are the cosines of the angles between the neutron's direction and the x and y axes, respectively. In the above equation, the source has been represented simply as S, but it must be remembered that this source includes scattering and fission terms that involve integrals of the neutron flux ψ over the angular variables μ and η. The treatment of these terms of the transport equation is well understood and not particularly dependent upon the form of the space mesh.

The discrete ordinate approximation to Eq. (1) consists of the set of equations

$$\mu_m \frac{\partial \psi_m}{\partial x} + \eta_m \frac{\partial \psi_m}{\partial y} + \sigma \psi_m = S_m, \quad m = 1, 2, \ldots, M, \qquad (2)$$

where the continuous variables μ and η have been replaced by the discrete points μ_m and η_m. The function $\psi_m(x,y)$ is then an approximation to the exact solution $\psi(x,y,\mu_m,\eta_m)$ in the m'th direction. The M equations above are coupled only through the source term S.

A number of difference approximations to Eq. (2) have been suggested. A currently popular finite-element method assumes that the flux $\psi_m(x,y)$ is linear in each triangle and is determined by the flux values at the vertices of the triangles. A difference equation for the unknown flux at each of these node points is then derived by multiplying by a weight function and integrating over the hexagonal region composed of the six adjacent triangles. With a proper choice of weight functions, this procedure is a Galerkin method and is equivalent to the minimization of a functional over the trial space of functions of the above form. It is, therefore, certain to give accurate answers and converge to the exact solution in the limit of an infinitely fine mesh. It is not clear, however, that such methods preserve neutron balance, in the sense that the total leakage plus absorption must equal the total source. Present difference schemes on orthogonal grids do preserve such balance, and this fact is utilized throughout current transport codes. A further disadvantage to using the finite-element methods is the dense coupling present in the equations that must be solved for the fluxes at the nodes. Each node point is coupled to the six adjacent nodes. Physically, the flux is coupled to only two of these nodes. The increased coupling in the difference equations means that they cannot be solved in a direct fashion by a single sweep through the mesh, as can be done with present schemes. An iterative procedure must be devised to solve such equations. Although it is not clear that this iteration will be slower than the source or inner iteration already present in transport codes, such an iterative process requires the storage of the complete angular flux. Present transport codes store only the scalar flux and enough moments of the angular flux to generate the scattering source. The complete angular flux contains 10 to 100 times as many numbers as the scalar flux for most problems. The additional storage required for this array exceeds the core or extended core capabilities of all computers, so that disk storage must be used. An iterative procedure involving repeated use of disk storage is likely to consume too much computer time.

For these reasons we abandon the finite-element schemes and search for methods similar to those now in use. One such method can be derived in the following manner. First, we introduce unknowns at the centers of the triangles and on the faces between triangles, in addition to the unknowns at the vertices of the triangles. Such a mesh arrangement is illustrated in Fig. 7.

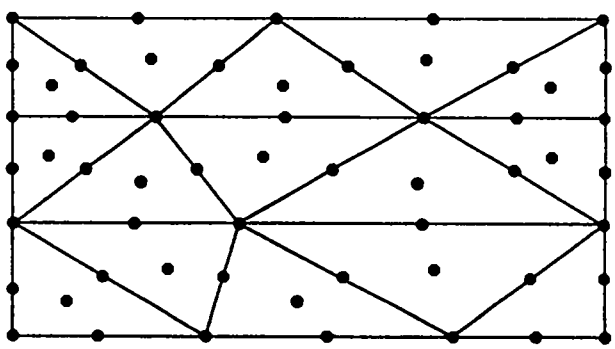


Fig. 7. Arrangement of unknowns on a triangular mesh.

With an even total number of triangles N, the total number of unknowns in a mesh like that of Fig. 7 is precisely 3N. There are $3N \pm 1$ unknowns for the m'th ordinate for an odd number of triangles. We must therefore derive a set of 3N or $3N \pm 1$ equations, depending on whether the mesh is odd or even. The first N equations are derived by integrating the discrete ordinate equation [Eq. (2)] over each of the N triangles.

Writing Eq. (2) as

$$\overline{\nabla} \cdot \hat{\Omega}_m \psi + \sigma\psi = S,$$

we have

$$\int_s \psi\, \hat{\Omega}_m \cdot \hat{n}\, ds + \sigma \int_V \psi\, d\tau = \int_V S\, d\tau,$$

where we have used the divergence theorem to express the volume integral as a surface integral for the first term. V is the volume of triangle being considered, and $\hat{n}$ is an outward pointing vector normal to the surface of the triangle. We can rewrite the above equation in the following form where the surface integral has been expressed as the sum of contributions from the three faces of the triangle.

$$\hat{\Omega}_m \cdot \hat{n}_1 \int_{s_1} \psi\, ds + \hat{\Omega}_m \cdot \hat{n}_2 \int_{s_2} \psi\, ds + \hat{\Omega}_m \cdot \hat{n}_3 \int_{s_3} \psi\, ds + \sigma \int_V \psi\, d\tau = \int_V S\, d\tau,$$

where $\hat{n}_1$, $\hat{n}_2$, and $\hat{n}_3$ are unit outward normals. We make the following definitions

$$\psi_1 = \frac{\int_{s_1} \psi\, ds}{s_1}, \tag{3a}$$

$$\psi_2 = \frac{\int_{s_2} \psi\, ds}{s_2}, \tag{3b}$$

$$\psi_3 = \frac{\int_{s_3} \psi\, ds}{s_3}, \tag{3c}$$

$$\psi_0 = \frac{\int_V \psi\, d\tau}{V}, \tag{3d}$$

and

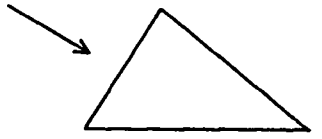
$$S_o = \frac{\int_V S\, d\tau}{V}, \tag{3e}$$

to obtain

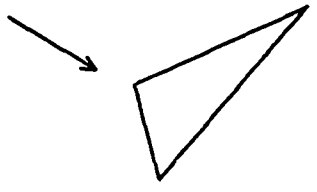
$$(\hat{\Omega}_m \cdot \hat{n}_1 s_1)\psi_1 + (\hat{\Omega}_m \cdot \hat{n}_2 s_2)\psi_2 + (\hat{\Omega}_m \cdot \hat{n}_3 s_3)\psi_3 + \sigma\psi_0 = S_o. \tag{4}$$

If we identify the fluxes ψ_1, ψ_2, and ψ_3 with the cell-face unknowns and ψ_0 with cell-centered unknowns, we obtain the first N difference equations. Equation (4) is a balance equation for a single cell and equates flow in minus flow out plus absorption to the cell source S_o.

An additional N equations can be obtained by assuming that the cell-centered fluxes $\psi_{i,j}$ are averages of either the cell-face fluxes or the cell-vertex fluxes. This leaves N or $N \pm 1$ equations needed to solve for the 3N or $3N \pm 1$ unknowns that we have introduced. To obtain these equations, we examine the two possible orientations of a triangle with respect to a single direction, as illustrated in Fig. 8. Triangles with the first orientation have only a single face visible from the specified direction; triangles with the second orientation have two faces visible. We assume that the neutron flux on the faces visible from a specified direction are known from boundary conditions



Orientation 1



Orientation 2

Fig. 8. The two possible orientations of a triangle with respect to a single direction.

or from previous calculations in adjoining cells. In this case, there are four unknown fluxes in triangles of Orientation 1 and two unknown fluxes in triangles of Orientation 2. Because we have already derived two equations per triangle, the two unknowns in triangles with the second orientation may be solved for immediately. If the triangle is of the first orientation, two additional equations are needed. They are obtained by assuming in these triangles that the two unknown cell-face fluxes are averages of the appropriate cell-vertex fluxes.

The above assumptions can be shown to yield precisely the needed number of equations. Furthermore, this set of equations may be solved in a single sweep through the mesh, provided the source is known. The order in which the unknowns are determined is slightly more complicated than for an orthogonal mesh. A simple example will best clarify this process. Consider a direction $\hat{\Omega}_m$ such that $\mu_m > 0$ and $\eta_m > 0$. The flux at points along the left and bottom edges of the system are known from boundary conditions. Using these boundary fluxes, the unknowns in the bottom-most band of triangles can be determined. The solution process for the bottom band can then be repeated successively to determine the fluxes in higher bands. We therefore consider only a single band of triangles as shown in Fig. 9. The first triangle is of Orientation 2, and the fluxes on the face between the first and second triangles are needed to solve for the flux in the first triangle. We must therefore skip the first triangle and solve for the fluxes in the second triangle, which is of Orientation 1. This determines the fluxes on the face between triangles 1 and 2, so that the fluxes in triangle 1 may now be determined. We must now skip the next three triangles to obtain a triangle in which sufficient boundary conditions are known to permit solution. The full solution process is illustrated in Fig. 10.

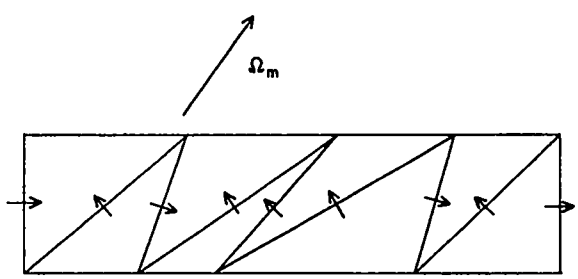


Fig. 9. A typical band of triangles with direction of flow across faces indicated by arrows.

Difference equations derived in the above manner have several attractive features. Equation (4) guarantees that neutron balance is retained, in the sense specified earlier. The difference equations may be solved by a direct sweep through the mesh, if the source is known, and it is not necessary to store the complete angular flux. Standard convergence acceleration devices, such as coarse mesh rebalancing, can be used for the source iterations with little modification. These facts make for a relatively easy introduction of such a difference scheme into present transport codes.

Let us now consider some of the details of the scheme we have just proposed. The form of the coefficients $\hat{\Omega}_m \cdot \hat{n}s$ appearing in Eq. (4) can be derived in the following manner. Assume that the projection of the vector $\hat{\Omega}_m$ onto the x and y axes is given by μ_m and η_m, respectively. The projection κ_m of $\hat{\Omega}_m$ on x' is determined as shown in Fig. 11.

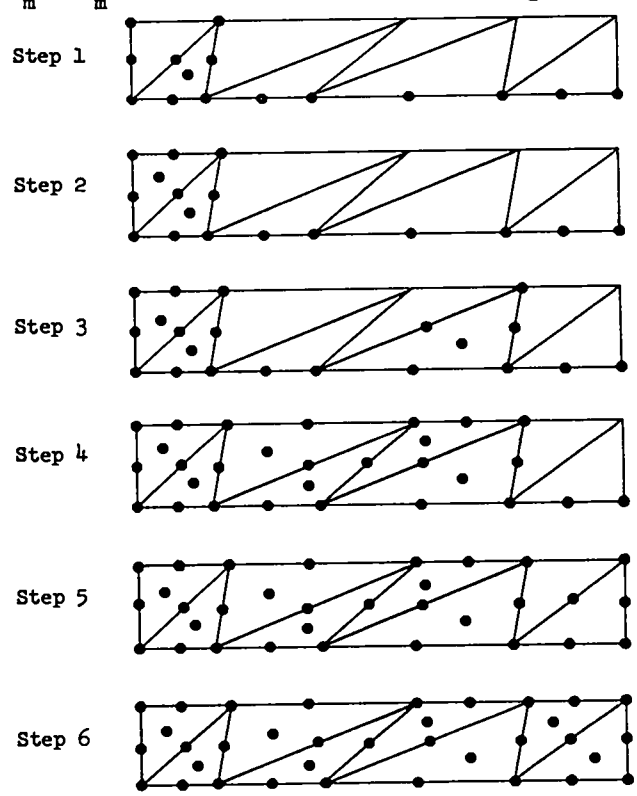


Fig. 10. A sample solution process on the band of triangles of Fig. 9. Dots represent known or determined fluxes.

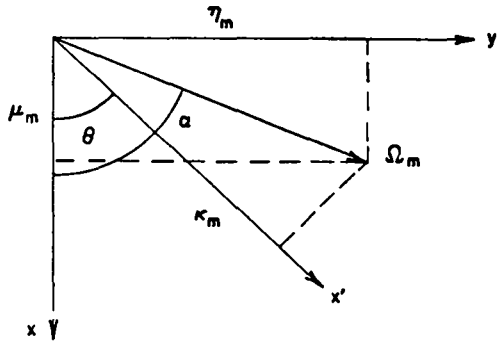


Fig. 11. Geometry for determination of κ_m.

We have

$$\cos \alpha = \mu_m / \sqrt{\mu_m^2 + \eta_m^2}$$

and

$$\sin \alpha = \eta_m / \sqrt{\mu_m^2 + \eta_m^2} .$$

However,

$$\kappa_m = \sqrt{\mu_m^2 + \eta_m^2} \cos (\alpha - \theta)$$

$$= \sqrt{\mu_m^2 + \eta_m^2} (\cos \alpha \cos \theta + \sin \alpha \sin \theta)$$

$$= \sqrt{\mu_m^2 + \eta_m^2} \left(\frac{\mu_m \cos\theta}{\sqrt{\mu_m^2+\eta_m^2}} + \frac{\eta_m \sin \theta}{\sqrt{\mu_m^2+\eta_m^2}} \right)$$

and

$$\kappa_m = \mu_m \cos \theta + \eta_m \sin \theta.$$

Now consider neutrons streaming in the direction Ω_m. We want to determine whether neutrons flow to the left or to the right across a particular face of a triangle (Fig. 12). We let $\Delta x = x_\ell - x_u$, where x_ℓ is the x coordinate of the lower point and x_u is the x coordinate of the upper point (see Fig. 12) of the face. If $\hat{n}$ is the rightward pointing normal to the face, then we have

$$\hat{\Omega}_m \cdot \hat{n} = \mu_m \cos \theta + \eta_m \sin \theta,$$

and

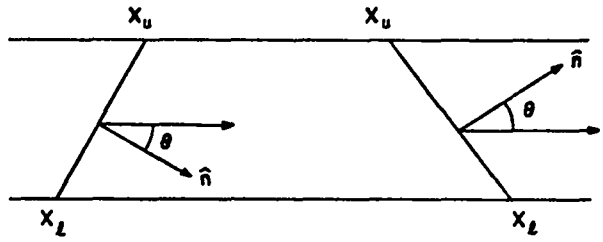


Fig. 12. Two triangle faces showing rightward normals $\hat{n}$.

$$\hat{\Omega}_m \cdot \hat{n} = \mu_m \frac{\Delta y}{\sqrt{\Delta y^2+\Delta x^2}} + \eta_m \frac{\Delta x}{\sqrt{\Delta y^2+\Delta x^2}} .$$

Let $s = \sqrt{\Delta x^2+\Delta y^2}$ be the area of the face of the triangle. Then we are actually interested in $\hat{\Omega}_m \cdot \hat{n}s$, which is given by

$$\hat{\Omega}_m \cdot \hat{n}s = \mu_m \Delta y + \eta_m \Delta x.$$

We note that the direction of flow is correctly predicted by the above formula in the special circumstances indicated in Table I.

TABLE I

DIRECTION OF FLOW IN CERTAIN CIRCUMSTANCES

	μ	η	Δx	FLOW
1	+	+	+	+
2	-	-	+	-
3	+	-	-	+
4	-	+	-	-

The situations in Table I are illustrated in Fig. 13.

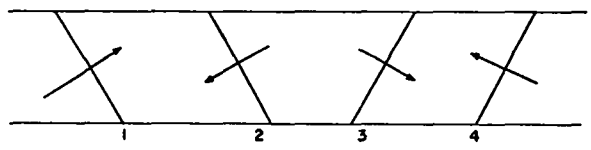


Fig. 13. Direction of flows given in Table I.

The coefficients $\hat{\Omega}_m \cdot \hat{n}$s appearing in Eq. (4) can now be written explicitly. We consider both upward and downward pointing triangles illustrated in Fig. 14. For the downward pointing triangle, these coefficients are given by

$$\hat{\Omega}_m \cdot \hat{n}_1 s_1 = \mu_m \Delta y + \eta_m (x_3 - x_1), \tag{5a}$$

$$\hat{\Omega}_m \cdot \hat{n}_2 s_2 = -\mu_m \Delta y - \eta_m (x_3 - x_2), \tag{5b}$$

and

$$\hat{\Omega}_m \cdot \hat{n}_3 s_3 = \eta_m (x_1 - x_2). \tag{5c}$$

The coefficients for the upward pointing triangles are

$$\hat{\Omega}_m \cdot \hat{n}_1 s_1 = \mu_m \Delta y + \eta_m (x_1 - x_3), \tag{5d}$$

$$\hat{\Omega}_m \cdot \hat{n}_2 s_2 = -\mu_m \Delta y - \eta_m (x_2 - x_3), \tag{5e}$$

and

$$\hat{\Omega}_m \cdot \hat{n}_3 s_3 = -\eta_m (x_1 - x_2). \tag{5f}$$

Let us now consider the solution of the set of difference equations for a particular triangle. We will assume initially that the cell-centered flux is the average of the cell-face fluxes; the other possibility will be considered later. If the triangle is of the second orientation, as depicted in Fig. 15, then we need only solve for ψ_0 and ψ_1 in terms of the boundary fluxes $b_1 - b_5$. We have the following equations.

$$A_1 \psi_1 - A_2 b_2 - A_3 b_4 + \sigma V \psi_0 = S_o V, \tag{6a}$$

and

$$\psi_1 + b_2 + b_4 = 3\psi_0, \tag{6b}$$

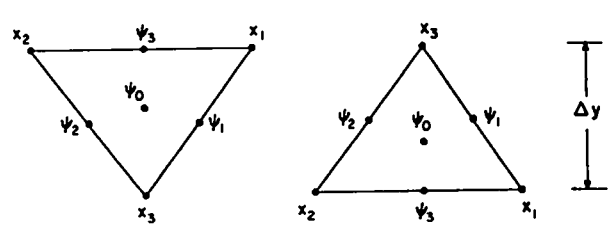


Fig. 14. Mesh definition for upward and downward pointing triangles.

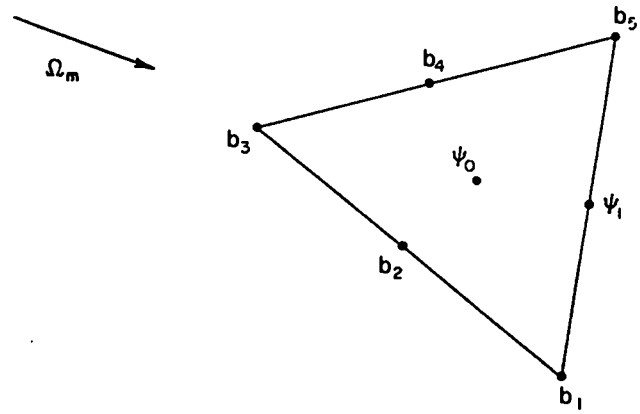


Fig. 15. A triangle of the second orientation.

where A_1, A_2, and A_3 are positive and equal in magnitude to $\hat{\Omega}_m \cdot \hat{n}_k s_k$, k = 1, 2, 3, in some order, depending on the position of the triangle. The solution of the above two equations is given by

$$\psi_0 = \frac{SV + (A_1 + A_2) b_2 + (A_1 + A_3) b_4}{3A_1 + \sigma V} \tag{7}$$

for the cell-centered flux. The cell-face flux is then found from

$$\psi_1 = 3\psi_0 - b_2 - b_4. \tag{8}$$

We note that the cell-centered flux is always positive if the boundary fluxes b_2 and b_4 and source S_o are positive, but that Eq. (8) may produce a negative cell-face flux. A similar problem arises on an orthogonal grid, where a device known as a negative flux fixup is used to guarantee positivity. We propose to use the same device on the triangular grid. If a negative ψ_1 is detected, it is set to zero and the cell-centered flux ψ_0 is recalculated from Eq. (6a) to preserve neutron balance.

If the triangle is of the first orientation, as shown in Fig. 16, more unknowns are involved. We must solve for the four unknowns ψ_0, ψ_1, ψ_2, and ψ_3 in terms of the boundary values b_1, b_2, and b_3. The equations are

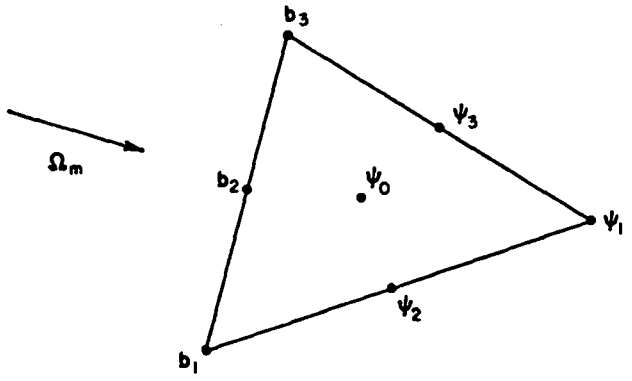


Fig. 16. A triangle of the first orientation.

$$A_1\psi_2 + A_2\psi_3 - A_3 b_2 + \sigma V\psi_0 = S_o V, \tag{9a}$$

$$3\psi_0 = \psi_2 + \psi_3 + b_2, \tag{9b}$$

$$2\psi_2 = b_1 + \psi_1, \tag{9c}$$

and

$$2\psi_3 = b_3 + \psi_1. \tag{9d}$$

The coefficients A_1, A_2, and A_3 are again positive and equal in magnitude to $\hat{\Omega}_m \cdot \hat{n}_k a_k$, $k = 1, 2, 3$, in some order. We solve for the cell-centered flux ψ_0

$$\psi_0 = \frac{S_o V + \left(\frac{A_2 - A_1}{4}\right) b_1 + \left(\frac{A_1 - A_2}{4}\right) b_3 + \left(A_3 + \frac{A_2}{2} + \frac{A_1}{2}\right) b_2}{\frac{3A_1}{2} + \frac{3A_2}{2} + \sigma V}. \tag{10}$$

Because the coefficient of either b_1 or b_3 is negative, positivity of the cell-centered flux cannot be guaranteed for this scheme. It is not known how serious this problem would become in practice. When ψ_0 is determined, the cell-vertex flux ψ_1 can be calculated from

$$\psi_1 = 3\psi_0 - \frac{b_1}{2} - \frac{b_3}{2} - b_2, \tag{11}$$

and the two cell-face fluxes can, in turn, be calculated from

$$\psi_2 = \frac{b_1 + \psi_1}{2}, \tag{12a}$$

and

$$\psi_3 = \frac{b_3 + \psi_1}{3}. \tag{12b}$$

Again, the use of a negative flux fixup is suggested. There are now four fluxes that may be negative in any combination, therefore such a scheme is likely to be complicated. The simplest but least accurate remedy is the replacement of Eqs. (9b) through (9d) by the following relation

$$\psi_0 = \psi_1 = \psi_2 = \psi_3. \tag{13}$$

The above assumption is similar to that of the "step" scheme on an orthogonal grid. It gives

$$\psi_0 = \psi_1 = \psi_2 = \psi_3 = \frac{S_o V + A_3 b_2}{A_1 + A_2 + \sigma V}, \tag{14}$$

and thus guarantees positivity of all cell fluxes.

In the above analysis, it was assumed that the cell-centered flux is the average of the cell-face fluxes. An equally valid assumption is that the cell-centered flux is the average of the cell-vertex fluxes. Slightly different results are obtained with this second assumption. For triangles with the second orientation, we replace Eq. (6b) with

$$3\psi_0 = b_3 + b_1 + b_5, \tag{15}$$

which completely determines ψ_0. The cell-face flux ψ_1 is then found from Eq. (6a) to be

$$\psi_1 = \frac{1}{A_1}\left[S_o V + A_2 b_2 + A_3 b_4 - \frac{\sigma V}{3}(b_3 + b_1 + b_5)\right]. \tag{16}$$

A fixup will again be needed because ψ_1 can be negative.

For triangles of the first orientation, we replace Eq. (9b) with

$$3\psi_0 = \psi_1 + b_1 + b_3. \tag{17}$$

Solving Eqs. (9a), (17), (9c), and (9d) for ψ_0, we obtain

$$\psi_0 = \frac{S_0 V + A_3 b_2 + \frac{A_1}{2} b_3 + \frac{A_2}{2} b_1}{\frac{3A_1}{2} + \frac{3A_2}{2} + \sigma V} . \qquad (18)$$

In this case, the cell-centered flux is always positive, but ψ_1 is given by

$$\psi_1 = 3\psi_0 - b_1 - b_3,$$

and may be negative. The fluxes ψ_2 and ψ_3 are again given by Eqs. (12) and may also be negative. The fixup scheme of Eqs. (13) and (14) can again be used to ensure positivity of the flux.

We next examine the truncation error of the above approximations to the discrete ordinate equations. To determine this truncation error, we substitute an exact solution of the discrete ordinate equations in the difference equations. The amount by which this exact solution fails to satisfy the difference equations is called truncation error.

Since Eq. (4) was derived by integrating the discrete ordinate equations over a triangular cell, an exact solution of the discrete ordinate equations satisfies this equation with no truncation error, provided the definitions [Eq. (3)] of ψ_0 - ψ_3 are used. We therefore need only examine the auxiliary assumptions we have made, which appear above as Eqs. (6b), (9b) through (9d), (15), and (17). With the notation of Fig. 17 we need only examine the following three equations, which are representative of those listed above.

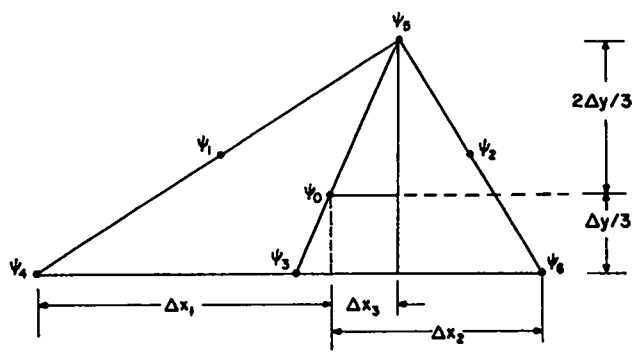


Fig. 17. Mesh definition for truncation error analysis.

$$\psi_1 = \frac{\psi_4 + \psi_5}{2}, \qquad (19a)$$

$$3\psi_0 = \psi_4 + \psi_5 + \psi_6, \qquad (19b)$$

and

$$3\psi_0 = \psi_1 + \psi_2 + \psi_3. \qquad (19c)$$

Recalling the definition [Eq. (3)] of ψ_1, ψ_2, and ψ_3, we have the following expression for the truncation error E_a of Eq. (19a)

$$E_a = \int_{s_1} \frac{\psi ds}{s_1} - \frac{\psi(4) + \psi(5)}{2},$$

where the integral is taken along the face between points 4 and 5 and $\psi(4)$ and $\psi(5)$ are the exact fluxes at points 4 and 5. The length of this side of the triangle is s_1. Expanding ψ about the point 1 we obtain

$$E_a = \frac{1}{s_1} \int_s \left[\psi + s\psi_s + \frac{s^2}{2}\psi_{ss} + \ldots\right] ds$$

$$- \frac{1}{2}\left[\left(\psi - \frac{s_1}{2}\psi_s + \frac{s_1^2}{8}\psi_{ss}\right)\right.$$

$$\left. + \left(\psi + \frac{s_1}{2}\psi_s + \frac{s_1^2}{8}\psi_{ss}\right) + \ldots\right]$$

$$= \psi + \frac{s_1^2\psi_{ss}}{24} - \psi - \frac{s_1^2\psi_{ss}}{8} + \ldots ,$$

and

$$E_a = -\frac{s_1^2\psi_{ss}}{12} + \ldots .$$

Thus, the truncation error of Eq. (19a) is of the order of the square of the length of the triangle face.

We next examine the truncation error E_b of Eq. (19b)

$$E_b = 3\psi(0) - \psi(4) - \psi(5) - \psi(6).$$

Expanding about the point 0,

$$E_b = 3\psi - [\psi - \Delta x_1 \psi x - \frac{\Delta y}{3} \psi y + O(\Delta x_1^2, \Delta y^2)]$$

$$- [\psi + \Delta x_3 \psi x + \frac{2\Delta y}{3} \psi y + O(\Delta x_3^2, \Delta y^2)]$$

$$- [\psi + \Delta x_2 \psi x - \frac{\Delta y}{3} \psi y + O(\Delta x_2^2, \Delta y^2)]$$

and

$$E_b = (\Delta x_1 - \Delta x_3 - \Delta x_2)\psi x + \left(\frac{\Delta y}{3} - \frac{2\Delta y}{3} + \frac{\Delta y}{3}\right)\psi y$$

$$+ O(\Delta x_1^2, \Delta x_2^2, \Delta x_3^2, \Delta y^2).$$

Consideration of the triangle in Fig. 17 shows that

$$\Delta x_2 + \Delta x_3 = \Delta x_1,$$

so we have

$$E_b = O(\Delta x_1^2, \Delta x_2^2, \Delta x_3^2, \Delta y^2).$$

With some additional algebra, the above error can be given by

$$E_b = O(s_1^2, s_2^2, s_3^2),$$

so that again the error is of second order.

The error of Eq. (19c) can be shown to be of second order by using previous results. We have

$$E_c = 3\psi(0) - \psi(1) - \psi(2) - \psi(3).$$

But we have shown that

$$\psi(1) = \frac{\psi(4) + \psi(5)}{2} + O(s_1^2).$$

In a similar fashion,

$$\psi(2) = \frac{\psi(5) + \psi(6)}{2} + O(s_2^2),$$

and

$$\psi(3) = \frac{\psi(4) + \psi(6)}{2} + O(s_3^2).$$

Substituting,

$$E_c = 3\psi(0) - \psi(4) - \psi(5) - \psi(6) + O(s_1^2, s_2^2, s_3^2)$$

$$= E_b + O(s_1^2, s_2^2, s_3^2),$$

and

$$E_c = O(s_1^2, s_2^2, s_3^2).$$

We have shown that all truncation errors are of second order. If the equations are stable (they have not been shown to be stable), then the true error, defined as the exact minus approximate solution, will be second order. Thus, the above methods should retain the accuracy characteristic of the diamond difference scheme on an orthogonal mesh. We note that the fixup routines suggested to ensure positivity of the flux are not, in general, second order. The use of these fixup routines may lead to a global loss of accuracy.

REFERENCES

1. B. G. Carlson and K. D. Lathrop, "Transport Theory - The Method of Discrete Ordinates," Chapter III of Computing Methods in Reactor Physics, (Greenspan, Kelber, and Okrent, Eds., Gordon and Breach, New York, 1968).

2. W. W. Engle, Jr., "A Users Manual for ANISN," K-1693, Union Carbide Corporation (1967).

3. K. D. Lathrop, "DTF-IV, a FORTRAN-IV Program for Solving the Multigroup Transport Equation with Anisotropic Scattering," LA-3373, Los Alamos Scientific Laboratory (1965).

4. K. D. Lathrop and F. W. Brinkley, "Theory and Use of the General-Geometry TWOTRAN Program," LA-4432, Los Alamos Scientific Laboratory (1970).