LA-UR-73-479

$2l,032$

TITLE: TRIANGULAR MESH METHODS FOR THE NEUTRON TRANSPORT EQUATION

AUTHOR(S): W. H. Reed and T. R. Hill

SUBMITTED TO: Proceedings of the American Nuclear Society

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

MASTER

TRIANGULAR MESH METHODS FOR THE

NEUTRON TRANSPORT EQUATION

by

Wm. H. Reed and T. R. Hill

University of California,
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87544

ABSTRACT

The methods that are developed in this paper for differencing the discrete ordinates equations on a triangular x-y grid are based on piecewise polynomial representations of the angular flux. The first class of methods discussed here assumes continuity of the angular flux across all triangle interfaces. A second class of methods, which is shown to be superior to the first class, allows the angular flux to be discontinuous across triangle boundaries. Numerical results illustrating the accuracy and stability of these methods are presented, and numerical comparisons between the above two classes of methods are made. The effectiveness of a fine mesh rebalance acceleration technique is also discussed.

I. INTRODUCTION

A two-dimensional (x,y) neutron transport code based on a triangular spatial mesh is currently under development at Los Alamos. This code will offer several advantages over present codes, all of which use an orthogonal mesh grid. By an orthogonal grid, we mean a grid in which all mesh lines meet at right angles. Many nuclear reactors are designed with hexagonal elements; these hexagonal geometries can be represented exactly with a triangular mesh by subdividing each hexagon into four, six, or more triangles. Furthermore, complicated curved geometries can be approximated easily and accurately with triangles.

The increased flexibility of a triangular mesh is not without added cost. Description of the mesh is more complicated, because the x and y coordinates of each vertex must be given. The order in which the mesh unknowns are solved is no longer straightforward but involves determining the direction of flow across triangle faces. Such determinations must be made repeatedly in the innermost iteration loops of a transport code, and they may increase computation times.

The purpose of this paper is to present some effective new schemes for obtaining finite-dimensional approximations to the transport equation on a triangular grid. Difference schemes for the transport equation fall into two broad categories, which we will refer to as implicit or explicit methods. In an implicit method no attempt is made to solve in the direction of the characteristics of the equation, that is, in the direction in which neutrons are streaming. Instead, variational methods 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. More properly stated, an explicit method follows characteristics through phase space. 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 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 characteristics. An implicit method couples all adjacent mesh cells with no regard for the direction of the characteristics. The diamond difference scheme is an explicit method; examples of implicit methods are given in Ref. 1.

Although both explicit and implicit methods have been studied thoroughly for rectangular meshes in x-y geometry, very few triangular mesh methods have been suggested, and even fewer have actually been tested. Ohnishi[2] proposes a Galerkin method with piecewise linear trial functions for the spatial

dependence of the flux coupled with a discrete ordinates treatment of the an-
gular variables, but he does not give numerical results supporting the method.
Several explicit methods are given in Ref. 3.

Very good results[4] have been obtained with implicit methods for relatively
small problems, that is, for problems in which the total number of cells in the
space-angle phase space is about 1000 or fewer. Unfortunately, many real phys-
ical problems involve such complicated geometries that several thousand spatial
mesh cells are needed to describe accurately the system boundaries and inter-
faces. With a relatively crude mesh for the angular variables, the total num-
ber of cells in phase space can be on the order of several tens of thousands.
It is not known whether implicit methods can solve problems of this size effi-
ciently. Furthermore, implicit methods require the storage of the complete an-
gular flux. Since each cell in phase space usually involves several unknown
function values, the number of storage locations required for the angular flux
for a single energy group can exceed 100,000. The storage required for a
100,000 by 100,000 matrix of coefficients can exceed one million locations,
assuming the matrix has a band width of about 10. These requirements exceed
the fast and extended core capacities of all modern computers, so that disc
storage must be used, even for a one-group problem. Present two dimensional
codes such as TWOTRAN[5] which are based on explicit methods store only the scalar
flux and enough moments of the angular flux to generate the scattering source.
Therefore, all parameters pertinent to a single energy group can usually be con-
tained in fast core, so that a more efficient program is obtained and data
transfer problems are minimized.

We are concerned in this paper with methods that are suitable for large
complicated physical problems. For the above reasons, it appears that explicit
methods may be superior to implicit methods for such problems. Thus we consid-
er only explicit methods in this paper.

Although the methods developed in this paper are applicable to a general
triangular mesh, we consider here only "regular" triangular meshes. A regular
triangular mesh is characterized by requiring that all vertices lie on horizon-
tal lines, so that horizontal bands of triangles are formed, and by insisting
that each interior vertex be common to six adjacent triangles. An example of
such a mesh is given in Fig. 1. Note that we do not require that triangles be
equilateral and that a non-rectangular domain is allowed.

There are two reasons why we consider only regular meshes. First, speci-
fication of a regular triangular mesh is much simpler than specification of a
general triangular mesh. Only three pieces of data are required: the mesh
spacings $(\Delta y)_j$, the x coordinates of the vertices along each horizontal line,
and the orientation of the first triangle on each band. The orientation of the
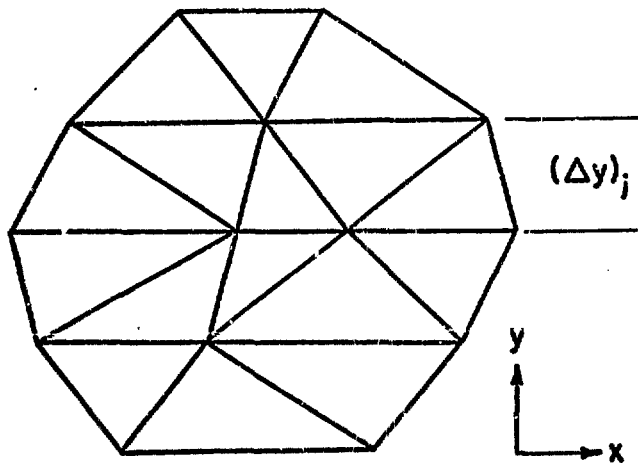
Fig. 1. A typical regular triangular mesh.

first triangle on each band can be specified by indicating whether the triangle
points up or points down. A regular triangular mesh is determined uniquely
by the above data.

The second reason why we consider only regular triangular meshes is relat-
ed to our decision to consider only explicit methods. Explicit methods neces-
sarily sweep the mesh in the direction of the characteristics, 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 loops of a transport code.
If complicated, such decisions would be prohibitively expensive. We restrict
to a regular triangular grid to simplify these decisions as much as possible
while retaining most of the flexibility of a general mesh.

II. THEORY

The one velocity neutron transport equation can be written in x-y geometry
as

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

where we have written the scattering, fission, and inhomogeneous sources simply
as S. In a multigroup context, S would also include sources due to scattering

and fission in other groups. We will utilize the standard discrete ordinates approximation to the above equation, thus we write

$$\mu_m \frac{\partial \psi_m}{\partial x} + \eta_m \frac{\partial \psi_m}{\partial y} + \sigma \psi_m(x,y) = S_m(x,y) \quad , \tag{2}$$

where the angular flux $\psi_m(x,y)$ is an approximation to $\psi(x,y,\mu_m,\eta_m)$ and a set of M quadrature points (μ_m,η_m) have been selected. For a detailed description of the standard quadratures used in two-dimensional discrete ordinates codes see Ref. 5. This reference also contains a good description of how the sources which we have written as $S_m(x,y)$ are generated. We assume the reader has a basic familiarity with standard discrete ordinates codes and take the liberty of omitting some of these details.

Our task is now the development of a discrete (in x and y) approximation to Eq. (2) on a triangular mesh. Since we consider only explicit methods, this task reduces to the problem of generating an approximation to $\psi_m(x,y)$ over a single triangle, assuming that $\psi_m(x,y)$ is known on the triangle boundaries visible when looking along the direction $\hat{\Omega}_m$ determined by μ_m and η_m. There are two cases that must be considered: one or two faces may be visible depending on the orientation of the triangle. These two cases are depicted in Fig. 2.

Orientation 1
(one face visible)
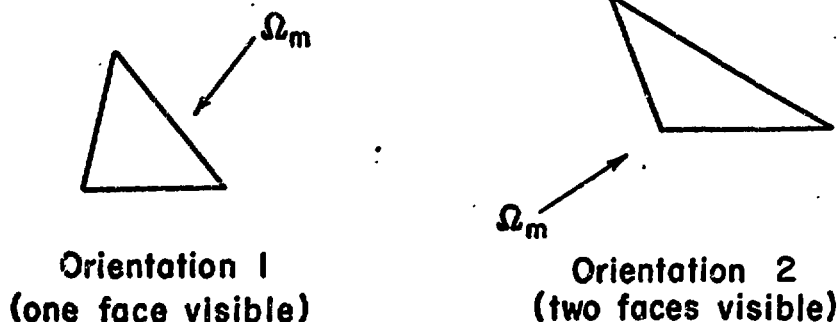
Orientation 2
(two faces visible)

Fig. 2. The two possible orientations of a triangle with respect to a direction $\hat{\Omega}_m$.

All methods developed in this paper assume that the angular flux over each triangle is given by a low-order polynomial. That is,

$$\tilde{\psi}_m(x,y) = \sum_{i=0}^{N} \sum_{j=0}^{N-1} A_{ij}^m x^i y^j \quad , \tag{3}$$

where $\tilde{\psi}_m \approx \psi_m$ over a given triangle and N is the order of the polynomial.

The form of the approximate solution $\tilde{\psi}_m$ over the entire system is completely determined by specifying the continuity conditions across triangle boundaries. The two methods considered in this paper differ only in the degree of continuity imposed on the approximate solution. The first method requires that the angular flux be continuous across all boundaries but does not require continuity of any derivatives of the solution. The second method imposes no continuity requirements whatsoever across triangle boundaries, that is, the angular flux is allowed to be discontinuous across all triangle boundaries. The flux on the boundary is to be the limit of the angular flux as one approaches the boundary in the direction Ω_m. The jump then occurs on the other side of the boundary.

We reiterate that the two methods considered in this paper are explicit methods which utilize piecewise polynomial representations of the angular flux. The order of the polynomials is arbitrary, and the effectiveness of higher order polynomials such as cubics and quartics is investigated numerically in the next section of this paper.

There are, of course, many ways in which a polynomial in x and y can be expressed. The representation of Eq. (3) is certainly the most common, but it is inconvenient for our purposes because the coefficients A_{ij} 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 $K \equiv \dfrac{(N+1)(N+2)}{2}$ distinct points (x_i, y_i) 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 $L_i(x,y)$ as the unique polynomial of order less than or equal N that is unity at the point (x_i, y_i) and is zero at the other $K - 1$ points. We refer to the K polynomials L_i so defined as Lagrange polynomials. If the points (x_i, y_i) have been chosen properly, then the Lagrange polynomials are linearly independent and form a basis for the space of polynomials of order less than or equal N. Thus we can replace Eq. (3) by the following equation with no loss of content:

$$\tilde{\psi}(x,y) = \sum_{i=1}^{K} \tilde{\psi}_i L_i(x,y) \quad , \tag{4}$$

where we have suppressed the subscript m. In the above equation, the coefficients $\tilde{\psi}_i$ can be interpreted as the value of $\tilde{\psi}(x,y)$ at the point (x_i, y_i), hence the notation $\tilde{\psi}_i$. It is this physical interpretation for $\tilde{\psi}_i$ which leads us to the Lagrange representation for $\tilde{\psi}(x,y)$.

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 points on each face of the triangle, with a point at each vertex. The remaining points are distributed uniformly in the interior of the triangle. Figure 3 illustrates the placement of these points for a few low order polynomials.
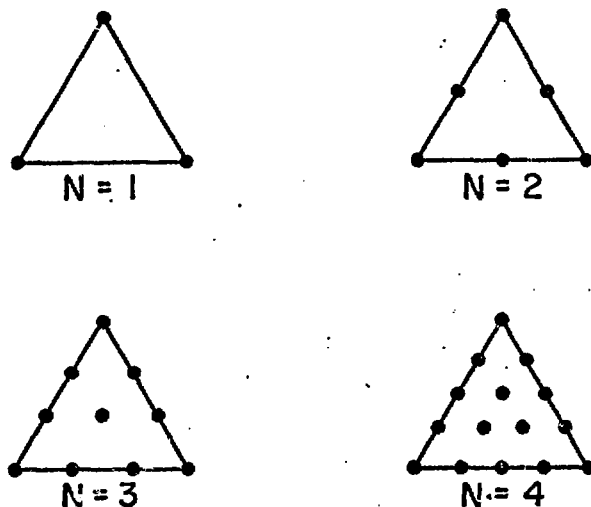

Fig. 3. The triangular point arrangement for a few low order polynomials.

Because a polynomial in x and y of order N is determined on a straight line uniquely by N+1 distinct points on the line, the boundary flux can be determined by the N+1 points on the boundary, without regard for the other points. Furthermore, the boundary flux is given by the unique one-dimensional polynomial which passes through these N+1 points.

We have now given a complete description of the form of the approximate solution for the two methods, continuous and discontinuous, that we discuss in this paper. What remains to be described is the manner in which this approximate solution is generated in each case. We consider first the continuous method.

The point arrangement indicated in Fig. 3 and the representation of Eq. (4) for $\widetilde{\psi}(x,y)$ over a triangle allow continuity to be imposed upon $\widetilde{\psi}(x,y)$ with little effort. We simply assume that $\widetilde{\psi}_i$ on all incoming boundaries of a triangle are known from prior calculation in adjacent cells or from system boundary data. An incoming boundary is a triangle boundary across which the neutron flow is into the cell. Of course, an incoming boundary for one cell is an outgoing boundary for the adjacent cell, and the definition of an incoming boundary depends upon the direction Ω_m under consideration. Since there can be one

or two incoming boundaries, either N+1 or 2N+1 of the coefficients $\tilde{\psi}_i$ of Eq. (4) are determined from continuity at the boundaries. This leaves a total of K - (N+1) or K - (2N+1) unknown coefficients $\tilde{\psi}_i$ per triangle, depending upon the orientation of Fig. 2. Let NN equal the number of unknowns in a given triangle. This parameter is tabulated in Table I for a few cases. We see from Table I that for linear polynomials and a triangle with two incoming boundaries there are no unknowns to be determined. We believe this situation to be undesirable and thus restrict our attention to polynomials of order greater than or equal two for the continuous method.

TABLE I

THE NUMBER OF UNKNOWNS NN IN A TRIANGLE
AS A FUNCTION OF ORIENTATION AND ORDER OF POLYNOMIAL

ORIENTATION	ORDER OF POLYNOMIAL	NN
1	1	1
1	2	3
1	3	6
2	1	0
2	2	1
2	3	3

We must now derive a set of NN equations for the NN unknowns on the given triangle. This is accomplished in the following manner. The assumed form of the solution is inserted in the discrete ordinate equation for the particular direction Ω_m under consideration. The resulting equation is then multiplied successively by each of a set of NN weight functions and integrated over the triangle. For the moment the weight functions are arbitrary and are denoted $W_u(x,y)$. With a proper choice of linearly independent weight functions, the above procedure gives the desired set of NN equations. This set of equations takes the form

$$\sum_{i=1}^{K} \left\{ \mu_m \left(W_j, \frac{\partial L_i}{\partial x} \right) + \eta_m \left(W_j, \frac{\partial L_i}{\partial y} \right) \right.$$

$$\left. + \sigma(W_j, L_i) \right\} \tilde{\psi}_i = (W_j, S_m) \quad , \tag{5}$$

$$j = 1, 2, \ldots NN \quad ,$$

where the inner product (a,b) represents the integral of ab over the triangle of interest. Note that some of the coefficients $\widetilde{\Psi}_i$ appearing on the left side of Eq. (5) are known from boundary data, so that in reality this equation represents an NN by NN linear algebraic system of equations for the unknown fluxes. With a proper choice of weight functions this system is nonsingular and can be solved routinely by any method appropriate for small linear systems, such as Gaussian elimination.

A good choice of weight functions is crucial to the success of the above method. We believe that the best weight functions are the polynomials of order less than or equal N-1 or N-2, depending upon whether the triangle is of orientation 1 or 2, respectively. One can easily verify that there are precisely NN of these polynomials in either case, so that we obtain the same number of weight functions as we have unknowns. Another possible choice of weight functions are the Lagrange polynomials that are unity at the unknown points. Numerical results in the next section indicate that the method does not perform as well with these Lagrange weights as with the low order polynomial weights. It is, of course, possible to choose weights so that the resulting method is unstable, in the sense that errors are amplified as one sweeps through the mesh. We have no theoretical results bearing on this problem, but we have never observed an instability with either of the above two choices of weights.

We consider next the second method in which the flux is allowed to be discontinuous across triangle boundaries. We again use the Lagrange representation of Eq. (4) for the flux and the point arrangement of Fig. 3. In this case, however, points lying on the triangle boundaries are thought of as actually lying in the interior of the triangle but arbitrarily close to the boundary. In this manner each boundary point splits into two or more points which are each associated with different triangles. We attempt to illustrate this point arrangement in Fig. 4. For this method the total number of unknowns is larger than that for the continuous method. It is clear that the number of unknowns per direction is in fact equal to K times the total number of triangles in the mesh.
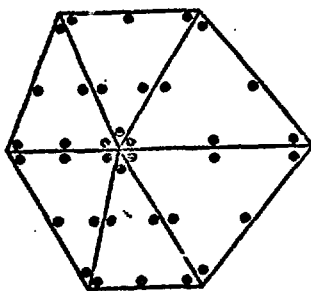

Fig. 4. A typical point arrangement for the discontinuous method. Boundary points are actually arbitrarily close to the boundary.

We now proceed in precisely the same manner as in method one. The assumed form of the solution in a given triangle is inserted in the transport equation, which gives a smooth function plus a Dirac delta function at the incoming boundaries due to the jump discontinuity at these boundaries. The resulting equation is multiplied by NN $\approx$ K weight functions and integrated over the triangle. Again we obtain an NN by NN linear algebraic system for the NN unknowns in each triangle, and with a proper choice of weight functions these equations are nonsingular and can be solved for the unknowns. Note that for this method the number of weight functions required is equal to the number of linearly independent polynomials of order less than or equal N. Our choice of basis is therefore immaterial, and any set of functions spanning the space of polynomials of order less than or equal N will give the same answer when used as weight functions. We have not investigated the use of more complicated non-polynomial weight functions for either of our methods. Again, we find experimentally that this discontinuous method is stable when polynomial weight functions are used.

III. NUMERICAL RESULTS

A one-group, isotropic scattering, discrete ordinates code was written to implement the methods of Sec. II. In this section, we present numerical results obtained with this code for several simple problems. An S_2 angular quadrature was used in all calculations.

The first test problem was designed to exhibit the accuracy that can be obtained with these methods. It consists of a one mean free path square containing a pure absorber. The source is isotropic and constant over the square, and boundary conditions are vacuum. Calculations were performed using the 200 triangle mesh of Fig. 5 and a similar 800 triangle mesh for both the continuous and discontinuous methods with the polynomial order N varying from one to four. Because we emphasize in this paper the spatial differencing of the transport equation, we choose to compare our computed results with the exact solution of the S_n equations, thus eliminating from consideration any errors introduced in the S_n approximation itself. The exact solution of the S_2 equations can be obtained easily for this simple homogeneous problem. In Table II we present the percentage difference between the total absorption computed from our numerical solutions and the total absorption computed from the exact S_2 solution. We note from the results of this table that the percent error decreases rapidly as the polynomial order is increased and that high order polynomial methods appear to be more efficient for obtaining answers accurate to many decimal places than low order polynomial methods.

We also note from the results of Table II that the high order polynomial methods are no more than second order accurate. This is seen in the following
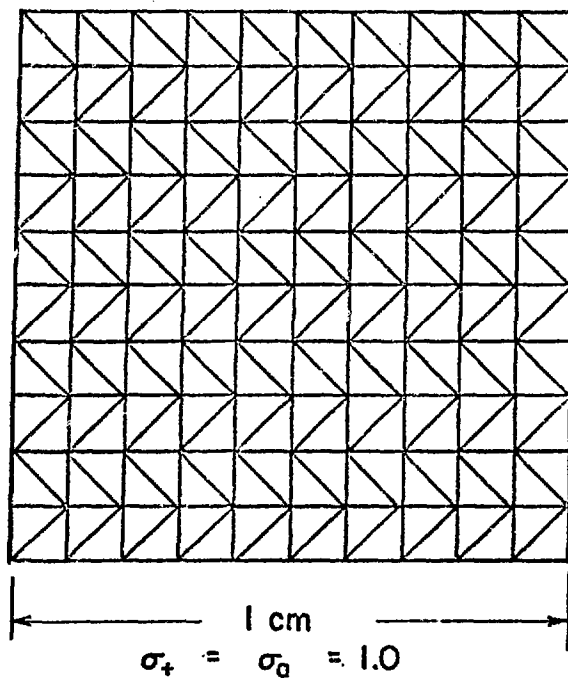
Fig. 5. Pure absorbing square 200 triangle mesh.

TABLE II

PERCENTAGE ERRORS IN TOTAL ABSORPTION
FOR A PIECEWISE POLYNOMIAL APPROXIMATION
TO THE SOLUTION OF THE TRANSPORT EQUATION
FOR A PURE ABSORBER IN A SQUARE WITH UNIFORM
SOURCE. WEIGHT FUNCTIONS ARE LOW ORDER POLYNOMIALS.

Number of Triangles	Method	Order of Polynomial	Computation Time (CDC-7600)	% Error
200	Continuous	2	0.52 sec	.005202
200	Continuous	3	1.50 sec	.000514
200	Continuous	4	4.12 sec	.000062
800	Continuous	2	1.98 sec	.001156
800	Continuous	3	5.45 sec	.000086
800	Continuous	4	13.57 sec	.000012
200	Discontinuous	1	.67 sec	.006565
200	Discontinuous	2	1.84 sec	.000294
200	Discontinuous	3	4.79 sec	.000177
200	Discontinuous	4	11.44 sec	.000044
800	Discontinuous	1	2.61 sec	.001330
800	Discontinuous	2	6.90 sec	.000065
800	Discontinuous	3	17.30 sec	.000027

manner. An increase in the number of triangles from 200 to 800 represents a halving of mesh spacings. We see from Table II for any order of polynomial that such a halving of mesh spacings yields about a factor of four reduction in the percent error. Thus all these methods are second order accurate in their predictions of total absorption rates. A closer examination of the flux shapes for this problem yields the result that these methods are in fact only first order accurate in their prediction of point values of the scalar flux but are second order accurate when predicting any integral parameter such as the total absorption or an eigenvalue.

The results of Table II were obtained using low order polynomials as weighting functions. Use of the Lagrange polynomials as weighting functions in the continuous method yields errors in the total absorption at least twice as large as those reported in Table II. For the discontinuous method, the choice of low order or Lagrange weights is immaterial.

The results of Table II do not indicate a clear superiority of either the continuous methods or the discontinuous methods. Although the discontinuous methods are somewhat more accurate for a given polynomial order than are the continuous methods, the latter utilize substantially fewer unknowns and require much less computation time. Nevertheless, we do believe that the discontinuous methods possess advantages which recommend their use. In particular, we find that the discontinuous methods are more stable than the continuous methods and that the acceleration method known as coarse mesh rebalance works better with the discontinuous methods. These claims will be substantiated by the next few test problems.

Transport theory methods based on continuous representations of the flux have great difficulty treating optically thick regions without using a fine mesh spacing. The diamond difference scheme can be derived by using a piecewise linear, continuous representation of the flux, and the tendency of this method to produce flux oscillations in such regions is well known. Transport codes based on this method always include some type of fixup scheme to eliminate these oscillations and the negative fluxes they produce, whenever possible. To examine the behavior of the discontinuous methods under such conditions, the first problem was repeated with a hundred-fold increase in the total cross section. The linear, discontinuous method, using the 200 triangle spatial mesh of Fig. 5, gave an error in the total absorption of 0.0027%. The TWØTRAN code (based on the continuous, diamond difference scheme), using a 100 square mesh, gave an error in the total absorption of 0.29%. The scalar flux along one half of the center plane is plotted in Fig. 6. The TWØTRAN solution is observed to oscillate about the infinite medium solution (0.01), whereas the discontinuous, triangular mesh solution rapidly damps to the infinite medium solution. The oscillation in the TWØTRAN solution would be more apparent if cell edge fluxes
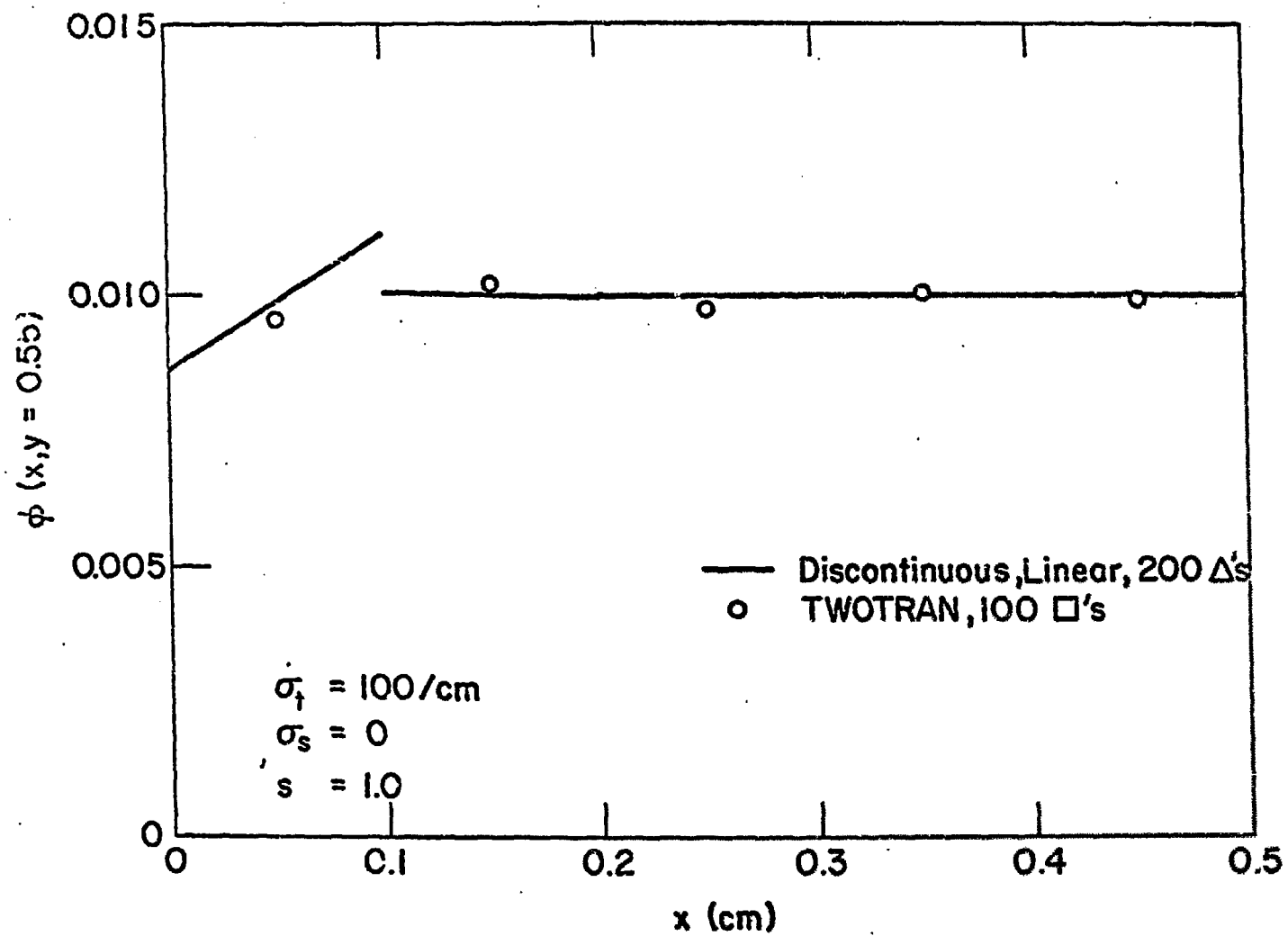
Fig. 6. Center plane scalar flux for Problem 2.

were plotted.

The stability of the discontinuous method is demonstrated again by Problem 3, diagrammed in Fig. 7. The triangular mesh calculations were performed with a 200 triangle mesh identical to Fig. 5. The TWØTRAN mesh consisted of 225 equally spaced squares. Scalar fluxes along one half of the center plane are plotted in Fig. 8. The continuous triangular mesh scheme exhibits large, slowly damped oscillations. Although the linear discontinuous method results in negative fluxes, they are relatively small in magnitude and rapidly damped. The negative flux fixup in TWØTRAN eliminates the difficulties of negative fluxes and oscillations in this case.

$$\sigma_t = 1.0 \quad \sigma_s = 0.95 \quad S = 1.0$$

$$\sigma_t = 100$$
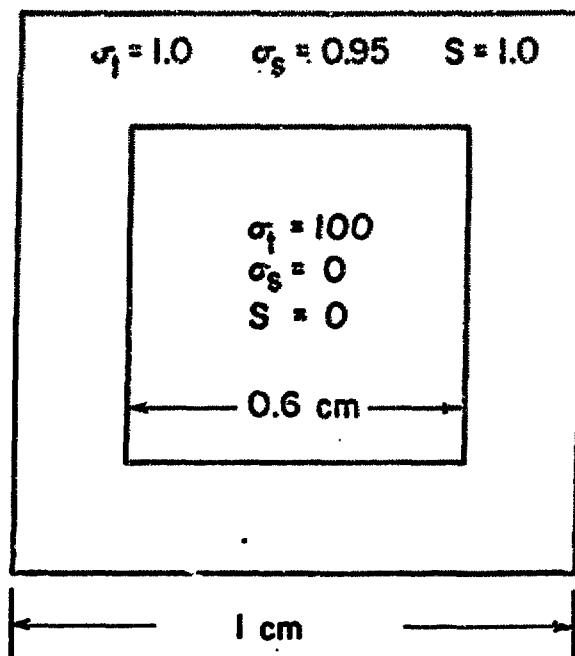$$\sigma_s = 0$$
$$S = 0$$

— 0.6 cm —

I cm

Fig. 7. Geometry for Problem 3.

The ability of a triangular mesh to treat curved boundaries accurately is illustrated by Problem 4, diagrammed in Fig. 9. The orthogonal TWØTRAN mesh of Fig. 10 gives a poor approximation to the curved boundary of the interior region. The triangular meshes shown in Figs. 11, 12, and 13 approximate the circular boundary in a much more accurate fashion. The total absorption for the various models is tabulated in Table III. The errors given are the errors in the absorption from the most accurate model, namely the 648 triangle mesh with the discontinuous, cubic difference scheme. We see that the continuous quadratic scheme gives significantly less accurate absorption rates than the
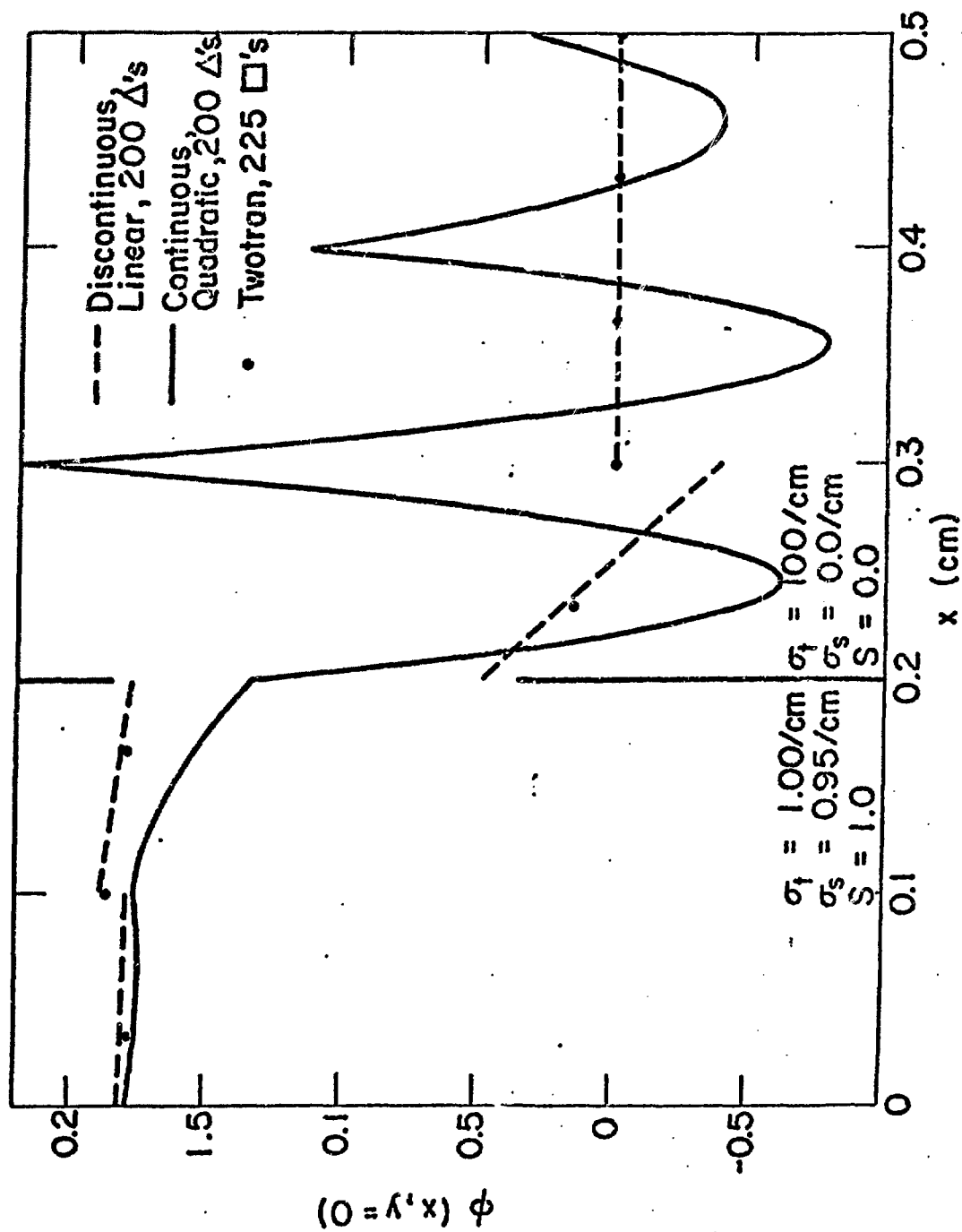
Fig. 8. Center plane scalar flux for Problem 3.

the discontinuous, linear scheme. The TWØTRAN square equivalent in Table III is the result for the problem in which the circular area is converted to a square of equal area. Both TWØTRAN results indicate the inability of a relatively coarse orthogonal grid to treat curved boundaries accurately.

Convergence of the inner or within-group iteration in a transport code can be slow if optically thick regions with scattering ratio near unity are present. In such situations the use of an acceleration technique is essential for reasonable computation times. One of the most effective acceleration methods is coarse mesh rebalance.[5] This method multiplies the fluxes in each coarse mesh zone by a factor for that zone chosen so that neutron balance over all zones is obtained. By neutron balance, we mean that for every zone the leakage plus absorption must equal the source. It is known that this acceleration can yield a divergent algorithm in some cases.[6] The convergence of the accelerated iteration appears to be related to the stability of the difference method, with the more stable schemes yielding the more rapidly convergent accelerated algorithms.[6] For this reason we expect our discontinuous methods to couple nicely with the rebalance acceleration technique to yield a rapidly convergent algorithm in almost all cases. The next problems are designed to test this hypothesis.

Problem 5 is a 10 mean free path square with a scattering ratio of 0.999, a unit source throughout the region, and vacuum boundary conditions. The linear discontinuous method and the 200 triangle mesh of Fig. 5 were used for the triangular mesh calculations. A 121 square mesh was used for the TWØTRAN calculations. The number of iterations and CDC-7600 computation time required for a point-wise flux convergence to 10^{-8} are given in Table IV for several rebalance schemes.

These schemes differ only in their definition of a coarse mesh zone. Each triangle is a separate coarse mesh zone in what we call fine mesh rebalance. In whole system rebalance the entire system comprises a single coarse mesh zone, and each band is a zone in band rebalance.

For the case of band or fine mesh rebalance, a linear algebraic system of equations must be solved for the rebalance factors. An iterative method is used to solve these equations, and ϵ_{rebal} is the convergence precision of these iterations. Since a tight convergence on the rebalance factors is unnecessary for the earlier inner iterations, a variable rebalance precision was examined in which ϵ_{rebal} was chosen as

$$\epsilon_{rebal} = 0.01 * \max_{i} |1 - f_i| \quad ,$$

with

$$10^{-1} \geq \epsilon_{rebal} \geq 10^{-8} \quad .$$

The f_i are the fine mesh rebalance factors from the previous inner iteration. An extrapolation procedure on the rebalance factors was also investigated whereby a corrected fine mesh rebalance factor is taken as

$$f_i^{corr} = \alpha(f_i - 1) + 1 \quad ,$$

$$0 \leq \alpha \leq 1 \quad .$$

Choice of $\alpha = 1$ corresponds to fine mesh rebalance and $\alpha = 0$ corresponds to no rebalance. An appropriately chosen α tends to dampen the oscillation of the rebalance factors from one inner iteration to the next.

Problem 6 is identical to Problem 5 except that the scattering ratio is unity and the square is 100 mean free paths wide. A comparison of the rebalance technqiues for this problem is shown in Table V.

Tables IV and V indicate that a large reduction in the number of inner iterations may result from the application of fine mesh rebalance. In particular, the gains appear to be much larger for the discontinuous difference schemes as opposed to the continuous difference scheme of TWØTRAN. For these problems the variable rebalance precision offers no savings in computation, whereas the extrapolation procedure effects a significant reduction in the number of inner iterations.
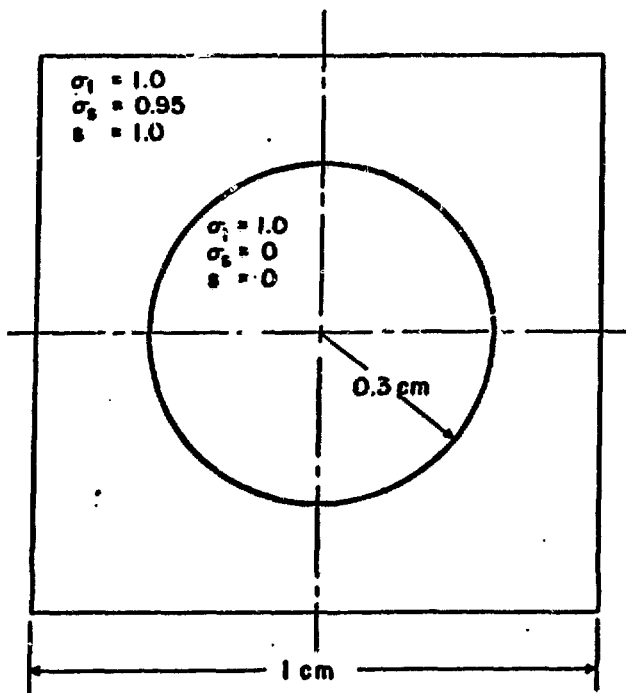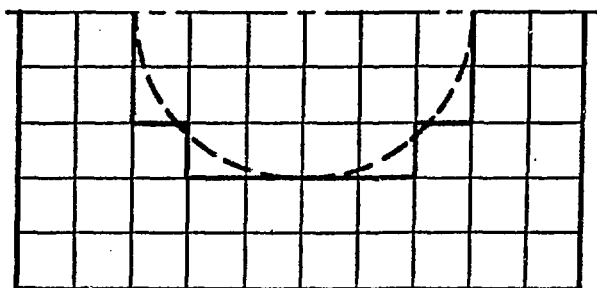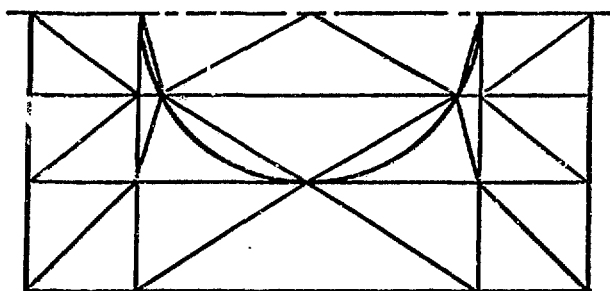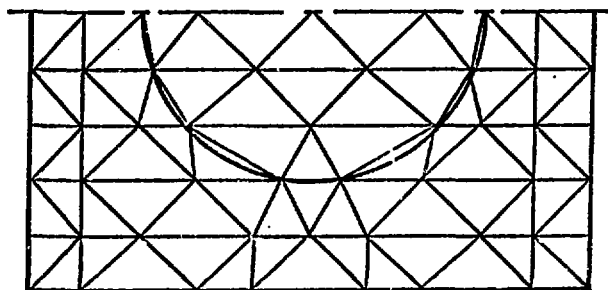
Fig. 9. Geometry for Problem 4.

TWOTRAN Mesh
100 cells

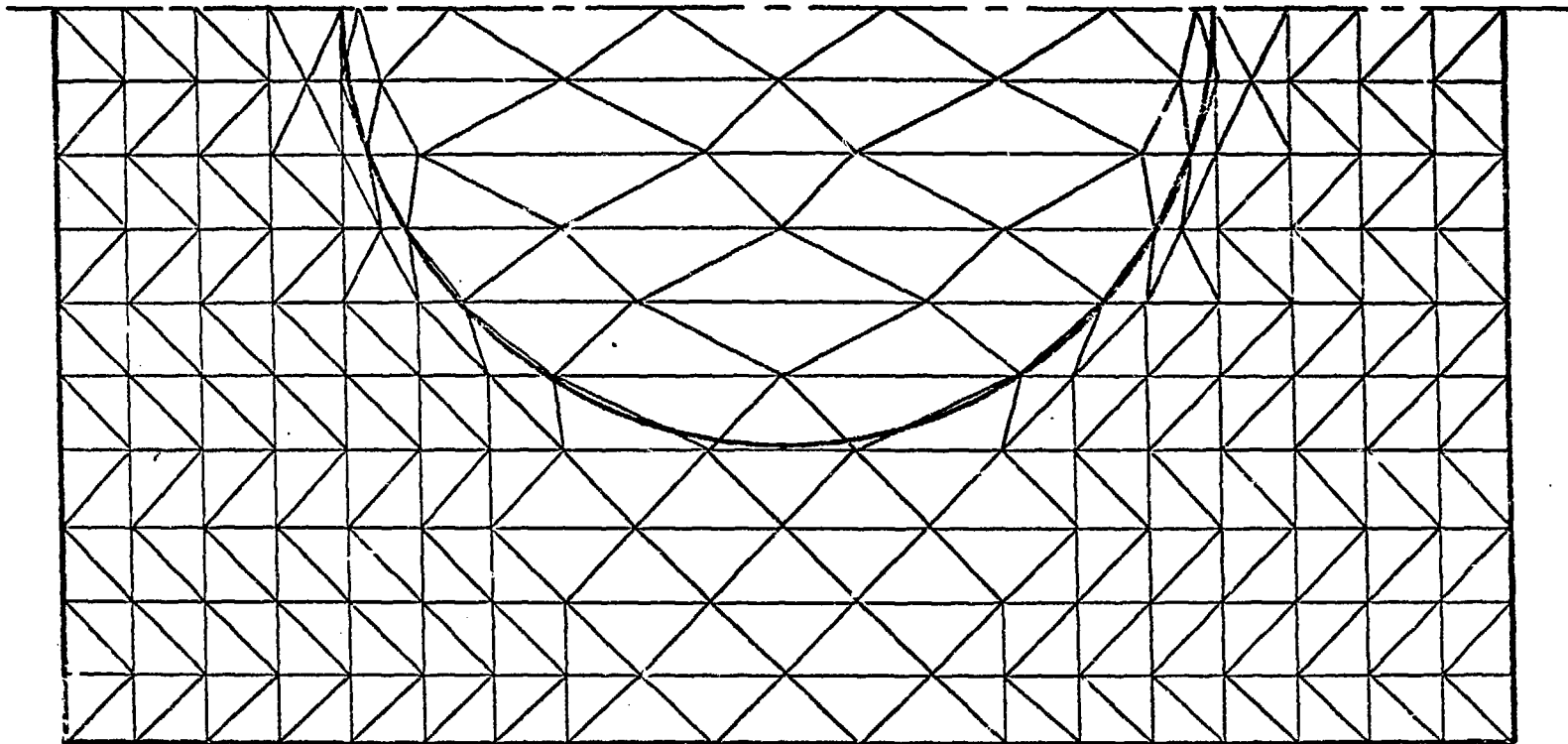Fig. 10. TWØTRAN mesh for Problem 4.

50 triangle mesh

Fig. 11.   Problem 4   50 triangle mesh

150 triangle mesh

Fig. 12.   Problem 4   150 triangle mesh

648 Triangle Mesh

Fig. 13. Problem 4   648 triangle mesh

TABLE III

TOTAL ABSORPTION FOR PROBLEM 4

MODEL	ABSORPTION		% ERROR
TWØTRAN, 100 square mesh	0.1064	3436	-0.282%
TWØTRAN Square equivalent, 1600 square mesh	0.1059	4559	-0.233%
50 triangle mesh, Continuous, Quadratic	0.1001	7172	+0.344%
150 triangle mesh, Continuous, Quadratic	0.1030	5362	+0.056%
648 triangle	0.1033	8129	+0.023%
50 triangle mesh, Discontinuous, Linear	0.1029	0304	+0.071%
150 triangle mesh, Discontinuous, Linear	0.1034	5378	+0.016%
648 triangle mesh, Discontinuous, Linear	0.1037	1294	-0.010%
648 triangle mesh, Discontinuous, Cubic	0.1036	1253	-

TABLE IV

INNER ITERATIONS REQUIRED FOR CONVERGENCE OF PROBLEM 5

ACCELERATION METHOD	ITERATIONS	COMPUTATION TIME (SEC)
TWØTRAN Mesh		
Whole system rebalance	82	1.96
Fine mesh rebalance	47	6.60
Fine mesh alternating with whole system rebalance	44	2.80
Triangular Mesh		
No rebalance	306	23.02
Whole system rebalance	87	6.72
Band rebalance	80	6.24
Fine mesh rebalance, $\varepsilon_{rebal} = 10^{-3}$	42	3.97
Fine mesh alternating with whole system rebalance	41	3.60
Fine mesh, variable ε_{rebal}	54	7.25
Fine mesh, variable ε_{rebal}, $\alpha = 0.70$	21	2.70

· TABLE V

INNER ITERATIONS REQUIRED FOR CONVERGENCE OF PROBLEM 6

ACCELERATION METHOD	ITERATIONS	COMPUTATION TIME (SEC)
TWØTRAN, Fine mesh alternating with whole system rebalance	> 1200	—
Triangular Mesh		
No rebalance	892	66.94
Whole system rebalance	455	34.32
Band rebalance	*	—
Fine mesh rebalance, $\epsilon_{rebal} = 10^{-3}$	71	7.99
Fine mesh alternating with whole system rebalance	75	6.90
Fine mesh rebalance, variable ϵ_{rebal}, $\alpha = 0.70$	75	12.39

*Iterations diverge

REFERENCES

1. William H. Reed, Nucl. Sci. Eng. 45, 309 (1971).

2. T. Ohnishi, "Application of Finite Element Solution Technique to Neutron Diffusion and Transport Equations," Proceedings, Conf. on New Developments in Reactor Mathematics and Applications, CONF-710302, Idaho Falls (1971).

3. Wm. H. Reed, "Triangular Mesh Difference Schemes for the Transport Equation," Los Alamos Scientific Laboratory report LA-4769 (1971).

4. W. F. Miller, Jr., E. E. Lewis, and E. C. Ressow, "The Application of Phase-Space Finite Elements to the Two-Dimensional Transport Equation in X-Y Geometry," submitted to Nucl. Sci. Eng. for publication.

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

6. W. H. Reed, "The Effectiveness of Acceleration Techniques for Iterative Methods in Transport Theory," Nucl. Sci. Eng. 45, 245 (1971).