J. King Saud Univ., Vol. 6, *Eng. Sci.* (1), pp. 53-77 (A.H.1414/1994)

ELECTRICAL ENGINEERING

"TDMGDSA" A Two Dimensional Multigroup Diffusion Synthetic Accelerated Transport Code

Tareq G. Aboalfaraj and Abdulghani M. Melaibari
Department of Nuclear Engineering
King Abdulaziz University, Jeddah, Saudi Arabia

(Received 3/12/1991; accepted for publication 18/5/1992)

Abstract. Neutron transport codes are essential tools in nuclear reactor and shielding design. Transport codes implement an iterative technique that suffers from slow convergence while solving highly scattering problems. This necessitated the development of acceleration methods, among which, the diffusion synthetic acceleration (DSA) is the fastest and the only stable method.

In this paper we present the development of "TDMGDSA", a two dimensional multi-group diffusion synthetic accelerated neutron transport computer code. TDMGDSA implements DSA to solve the discrete ordinate equations in X-Y geometry for three spatial differencing schemes; the diamond, weighted diamond, and linear discontinuous differencing. TDMGDSA is the first to implement a linear form of DSA for the diamond differencing, and is one of the first to implement DSA for the weighted diamond and linear discontinuous differencing schemes. TDMGDSA has been tested and is shown to be working reliably.

Introduction

Nuclear reactor engineering and design requires neutron transport computer codes that are fast, stable, and accurate. Typically, such codes involve solving the discrete-ordinates multigroup neutron transport equation in two or three dimensional geometries. Transport codes used for designing thermal reactors use iteration methods that often converge very slowly. Different methods have been developed to accelerate the convergence of these codes. The coarse mesh rebalance method [9] and the Chebychev acceleration method [1] are examples of earlier approaches to solve this problem.

Diffusion Synthetic Acceleration [2-11] (DSA) is a very powerful alternative acceleration method that is stable, regardless of the size of the spatial mesh, and is specially effective in highly scattering media.

The objective of this paper is two folds. *First,* to develop diffusion synthetic acceleration algorithms, based on the procedure developed by AboAlfaraj [13], that *take into account* the linearly anisotropic scattering and source terms for three spatial discretization schemes; namely the *diamond,* the *weighted diamond,* and the *linear discontinuous* schemes [10]. *Second,* to develop a two dimensional multigroup neutron transport code "TDMGDSA", that implements these algorithms to accelerate the convergence of the solution.

The Diffusion Synthetic Acceleration Method

The *synthetic* acceleration method was first proposed by Kopp [2] in 1963. Kopp introduced most of his basic ideas in the context of the one group neutron transport equation in slab geometry. The synthetic method involves the construction of the solution of a difficult problem by combining the solutions to a sequence of much easier problems that have the same solution. Crawford and Friedman [3] adopted Kopp's method to more general, multidimensional geometries, and used the diffusion equation to accelerate Monte Carlo transport computations. In 1969, Gelbard and Hageman [4] implemented Kopp's method for the discrete-ordinates transport equation. Their implementation is the basis of the current DSA methods, and hence will be described briefly here.

The one-energy neutron transport equation with isotropic source and scattering can be written in the form

$$\phi = K(\sigma_s \phi + S), \tag{1 a}$$

where σ_s is the scattering cross section, S is the neutron source density, and K is a form of the transport operator. Let us assume that K_H is an operator that approximates K very closely, while K_L is a cruder approximation of K. Then both

$$\phi_H = K_H(\sigma_s \phi_H + S) \tag{1 b}$$

$$\phi_L = K_L(\sigma_s \phi_L + S), \tag{1-c}$$

apptroximate the transport equation. Eq. (1-b) is called the "high order" (almost exact) equation, while Eq. (1-c) is called the "low order" equation. Typically, the high order equation is the discrete ordinates equation, while the low order equation is a diffusion-like equation. In practice, the solution we seek is the solution of the high order equation, which should approximate the transport solution well. The basic principle of the synthetic method is to use the low order equation to accelerate the convergence of the high order equation.

To show how this is done, let us assume that $\emptyset^{(0)}$ is some approximate solution to Eq. (1-b), and let

$$\phi_H = \phi^{(0)} + \varepsilon^{(0)}. \qquad (1-d)$$

An equation for the error, $\varepsilon^{(0)}$, can be obtained by substituting Eq. (1-d) into Eq. (1-b) to get

$$\phi^{(0)} + \varepsilon^{(0)} = K_H\left(\sigma_s\phi^{(0)} + \sigma_s\varepsilon^{(0)} + S\right), \qquad (1-e)$$

$$\left(I - K_H\sigma_s\right)\varepsilon^{(0)} = K_H\left(\sigma_s\phi^{(0)} + S\right) - \phi^{(0)} \equiv R^{(0)}, \qquad (1-f)$$

where $R^{(0)}$ is *residual* and I is the identity operator. Instead of the high order equation, the low order equation is used to estimate ε, through the equation

$$\varepsilon^{(0)} = \left(I - K_H\sigma_s\right)^{-1} R^{(0)} \approx \left(I - K_L\sigma_s\right)^{-1} R^{(0)} \qquad (1-g)$$

Hence, the transport problem is solved by doing the following two steps in each iteration

$$\phi^{(i+1/2)} = K_H\left(\sigma_s\phi^{(i)} + S\right), \qquad (1-h)$$

$$\phi^{(i+1)} = \phi^{(i+1/2)} + \left(I - K_L\sigma_s\right)^{-1}\left(\sigma^{(i+1/2)} - \phi^{(i)}\right) \qquad (1-i)$$

The first step is the usual transport sweep, but now followed by a correction step in which the low order operator is used. From Eq. (1-i) it is clear that we need to invert the low order operator. It might be possible for some cases, depending on the low order operator, to perform a direct inversion, but normally an iteration method is used.

In their work, Gelbard and Hageman used both the diffusion and S_2 "low order" operators to accelerate the "high order" transport equation. Their analysis showed that the iterative solution of the transport equation can be greatly accelerated by using the synthetic acceleration method. To confirm these findings they wrote two computer codes to compare the synthetic approach with the block Gauss-Seidel accelerated transport equation. The results showed that the synthetic approach was very effective for problems where the scattering ratio (σ_s/σ_t) is close to unity. However, Reed [5] then compared the synthetic method to the coarse mesh rebalance method and showed that both methods become unstable for optically thick $(\sigma_t \Delta x \gg 1)$ spatial meshes. Reed's analysis showed that the synthetic method should be applied to meshes that are not larger than about one neutron mean free path. This instability imposed a severe limitation that prevented the synthetic method from being used in practical problems.

The source of this instability was later explained by Alcouffe [6-8], who developed for the first time a stable diffusion synthetic acceleration method. Alcouffe showed that if the synthetic equations are derived *consistently* with the transport equation, a stable acceleration method, regardless of the size of the mesh, is obtained. Reed derived the analytic low order equation directly from the analytic transport equation, and then he spatially discretized both equations independently from one another. This created an *inconsistency* between the discretized transport and diffusion equations. For small meshes the inconsistency is less important, thus it allows the method to accelerate effectively. However, as the mesh size becomes large, the inconsistency grows and ultimately makes the method unstable. It is very important to note that the success of DSA is highly dependent on the use of a differencing scheme for the low order equation that is consistent with the differencing scheme for the transport equation. However, there is no a priori guarantee that consistency alone produces a stable method.

In the practical implementation of his DSA method, Alcouffe [8] introduced *inconsistent* linear approximations to the diffusion terms, and he non-linearly treated the removal term to manipulate the low order equations into a new form that can be more readily solved iteratively by the multigrid method. These manipulations have been suspected to be the cause of the degraded performance of Alcouffe's DSA form experienced in some problems [20].

Alcouffe's *consistency* ideas were later generalized by Larsen [9], who introduced a procedure that results in unconditionally stable acceleration for several discretization methods for the transport equation. Larsen used a separation of variables technique to Fourier analyze the stability of the resulting methods. This Fourier analysis technique has proved to be a very effective way to analyze different acceleration methods. Larsen tested his procedure on various differencing schemes in slab geometry, showing great improvement on the rate of convergence.

AboAlfaraj [13-14] showed that the particular form of the DSA method encoded in the TWODANT [18] computer code for solving neutron transport problems in X-Y geometry does exhibit poor performance as the spatial mesh size increases. He proposed an alternative form and showed that the new form produces an unconditionally stable acceleration that is much more efficient than the existing form especially for problems with large spatial meshes.

In two-dimensional geometries, the diamond-differenced transport equation is the only discretized transport equation for which the DSA procedure yields easily solvable low order equations. Other differencing schemes produce system of low order equations that are *algebraically* impossible to collapse into a solvable equation. AboAlfaraj [13] developed for the first time a method that produces a set of diffusion synthetic acceleration equations for the weighted diamond and the linear discontinu-

ous spatial discretization schemes in X-Y geometry. He showed that his method performs very well in typical reactor and neutron shielding problems characterized with spatial meshes smaller than four neutron mean free paths. For larger meshes, the method degrades but remains stable.

The Multigroup Transport Equation

The multigroup transport equation (10) can be written in the general form

$$\left[\hat{\Omega}.\hat{\nabla}+\sigma_g(r)\right]\psi_g\left(r,\hat{\Omega}\right)=\sum_{g'=1}^{G}\int d\Omega'\ \sigma_{gg'}\left(r,\hat{\Omega}.\hat{\Omega}'\right)\psi_{g'}$$

$$+\chi_g\sum_{g'=1}^{G}\nu\sigma_{fg'}(r)\phi_{g'}(r)+q_g^e\left(r,\hat{\Omega}\right). \qquad (2-a)$$

Here we used standard notation. The above equation can be written more compactly as,

$$L\psi\left(r,\hat{\Omega}\right)=S\psi\left(r,\hat{\Omega}\right)+q\left(r,\hat{\Omega}\right). \qquad (2-b)$$

The L operator represents the neutron loss due to leakage and absorption, the S operator represents the production of neutrons due to scattering, and the term q represents neutron production due to fission and external sources. The scattering term can be divided into three parts, upward scattering S_u, within-group. scattering S_s and downward scattering S_d. Thus,

$$S=S_u+S_s+S_d, \qquad (3)$$

Now if Eq. (3) is substituted into Eq. (2-a), we obtain,

$$\left(L-S_d-S_s\right)\psi=S_u\psi+q \qquad (4)$$

This is solved iteratively as follows:

$$\left(L-S_d-S_s\right)\psi^{(l+1)}=S_u\psi^{(l)}+q \qquad (5-a)$$

Equivalently,

$$\left[\hat{\Omega}.\hat{\nabla}+\sigma_g(r)\right]\Psi_g^{(l+1)}\left(r,\hat{\Omega}\right)-\sum_{g'=1}^{g}\int d\Omega'\sigma_{gg'}\left(r,\hat{\Omega}.\hat{\Omega}'\right)\Psi_g^{(l+1)}$$

$$=\sum_{g'=g+1}^{G}\int d\Omega'\sigma_{gg'}\left(r,\hat{\Omega}.\hat{\Omega}'\right)\psi_g^{(l)}+\chi_g\sum_{g'=1}^{G}V\sigma_{fg'}(r)\phi_g^{(l)}(r)+q_g^e\left(r,\hat{\Omega}\right) \qquad (5-b)$$

To understand how the multigroup transport equation is solved we need to closely examine Eq. (5-a) and (5-b). The first equation describes an outer iteration procedure. Starting with some given *multigroup* estimate of the neutron flux $\Psi^{(l)}$, the right had side of the first equation is computed. Using Eq. (5-b), the within-group flux is calculated for each energy group, starting with the highest energy (g = 1) and proceeding to the low energy group (g = G). This results in a new estimate for the flux $\Psi^{(l+1)}$, and the iteration is repeated until Ψ converges.

The solution of equation (5-b) involves an *inner* iteration which can be explained by rewriting the equation in the form

$$L_g\psi_g^{(l+1)}-S_{sg}\psi_g^{(l+1)}=S_g^{(l+1)} \qquad (6)$$

Here we have moved all references of other energy group fluxes to the right side, which are known at the beginning of each group calculation, and can be thought of as a fixed source. The left side of Eq. (6) contains two terms, the loss of neutrons due to leakage and absorption, L, and the production of neutrons due to self-group scattering, S_{sg}. The presence of the self scattering term makes it difficult to solve the above equation by directly inverting the left side operator. This necessitates solving the within group equation iteratively. Such a procedure, known as the *inner iteration*, can be described by

$$L_g\psi_g^{(l+1,k+1)}=S_{sg}\psi_g^{(l+1,k)}+S_g^{(l+1)} \qquad (7)$$

Here the k index refers to the k'th inner iteration.

The above within group inner iteration is actually a form of the fixed source iteration method. With the appropriate values for the source and the material cross sections, the problem can be reduced to a one group transport equation with fixed source.

It should be clear that the inner iteration constitutes the heart of each transport calculation. The computer spends most of the time performing these iterations for each energy group in each outer iteration. In order to solve the multigroup transport equation quickly, an acceleration technique is needed for rapidly solving the inner

iterations. The similarity between the within group equations and the one group fixed source transport equation is the primary reason behind the interest in accelerating source iteration problems. Acceleration techniques that perform well with the one-group fixed source problems can be quite easily extended to multigroup transport calculations.

Diffusion Synthetic Acceleration of the Fixed Source Discrete-Ordinates Equations in X-Y Geometry

Stable and reliable diffusion synthetic acceleration is used to quickly solve the fixed source problem "inner iteration" that constitutes the heart of the multigroup calculations. The forms of the Diffusion Synthetic Acceleration implemented by AboAlfaraj [13-14] is used here. Support for *linearly anisotropic* source and scattering were added resulting in somewhat more complicated equations but essentially with the same flavour. The detailed derivations are available [15], only the final equations are shown here.

In order to numerically solve the problem, the transport equation has to be descretized both in space and direction. The problem is divided into rectangular cells as shown in Fig. 1. Each cell is assumed to consist of a uniform material composition. Hence the cross sections are constant within each cell and are allowed to vary from cell to cell. Neutron direction of travel is also limited to fixed number of discrete ordinates.

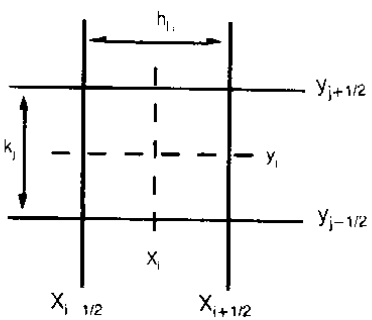

Fig. 1. X-Y geometry spatial cell

Starting with an initial flux, solving the fixed source problem consists of doing a transport sweep [10] optionally followed by a diffusion acceleration step. This acceleration generates correction factors *f*'s that are added to the corresponding transport flux. The new estimate is then used in the next iteration, the process is repeated until the flux converges.

The spatially descretized one group discrete-ordinates neutron transport equation with linearly anisotropic source and scattering in x-y geometry is given by [10].

$$\frac{\mu_m}{h_i}\left(\psi^{(l+1/2)}_{m,i+1/2,j}-\psi^{(l+1/2)}_{m,i-1/2,j}\right)+\frac{\eta_m}{k_j}\left(\psi^{(l+1/2)}_{m,i,j+1/2}-\psi^{(l+1/2)}_{m,i,j-1/2}\right)$$

$$+\sigma_{ij}\psi^{(l+1/2)}_{m,ij}=\sigma_{s00,ij}\phi^{(l)}_{00,ij}+3\mu_m\sigma_{s10,ij}\phi^{(l)}_{10,ij}+3\eta_m\sigma_{s1j}\phi^{l}_{00,ij}+S_{m,ij}, \qquad (8-a)$$

In order to relate the angular flux at the cell center to those on the cell edges, we introduce some *Auxillary Equations*. In current version of the code only *Diamond Differencing, Weighted Diamond* and *Linear Discontinuous* [13-15] are supported. The general form of the auxiliary equations is given by:

$$\psi^{(l+1/2)}_{m,ij}=\frac{1+\alpha_{m,ij}}{2}\psi^{(l+1/2)}_{m,i+1/2,j}+\frac{1+\alpha_{m,ij}}{2}\psi^{(l+1/2)}_{m,i-1/2,j}$$

$$-\Theta\alpha_{m,ij}\left(\frac{\hat{S}_{ij}+\sigma_{s00,ij}\hat{\phi}^{l}_{00,ij}+3\mu_m\sigma_{s10,ij}\phi^{l}_{10,ij}+3\eta_m\sigma_{s11,ij}\hat{\phi}^{l}_{11,ij}}{\sigma_{ij}}-\frac{\hat{\psi}^{(l+1/2)}_{m,i,j+1/2}-\hat{\psi}^{(l+1/2)}_{m,i,j-1/2}}{\tilde{\varepsilon}_{m,ij}}\right), \qquad (8-b)$$

$$\psi^{(l+1/2)}_{m,ij}=\frac{1+\beta_{m,ij}}{2}\psi^{(l+1/2)}_{m,i+1/2,j}+\frac{1+\beta_{m,ij}}{2}\psi^{(l+1/2)}_{m,i-1/2,j}$$

$$-\Theta\beta_{m,ij}\left(\frac{\tilde{S}_{ij}+\sigma_{s00,ij}\tilde{\phi}^{l}_{00,ij}+3\mu_m\sigma_{s10,ij}\tilde{\phi}^{l}_{10,ij}+3\eta_m\sigma_{s11,ij}\tilde{\phi}^{l}_{00,ij}}{\sigma_{ij}}-\frac{\tilde{\psi}^{(l+1/2)}_{m,i+1/2,j}-\tilde{\psi}^{(l+1/2)}_{m,i+1/2,j}}{\hat{\varepsilon}_{m,ij}}\right), \qquad (8-c)$$

The above form has the characteristic that it can be easily reduced to the diamond, weighted diamond or linear discontinous differencing forms by simply redefining the terms $\alpha_{m,ij}$, $\beta_{m,ij}$ and Θ such that,

$$\alpha_{m,ij} = \begin{cases} \dfrac{\hat{\varepsilon}_{m,ij}}{6+\left|\hat{\varepsilon}_{m,ij}\right|}, & LD \\ \alpha\dfrac{\mu_m}{|\mu_m|}, & WD, \\ 0, & DD \end{cases} \qquad (9-a)$$

$$\beta_{m,ij} = \begin{cases} \dfrac{\bar{\varepsilon}_{m,ij}}{6+\left|\bar{\varepsilon}_{m,ij}\right|}, & LD \\ \beta\dfrac{\eta_m}{|\eta_m|}, & WD, \\ 0, & DD \end{cases} \qquad (9-b)$$

$$\Theta = \begin{cases} 1, & LD \\ 0, & Otherwise \end{cases} \qquad (9-c)$$

All new terms appearing in the above and following equations are defined in appendix A. The spherical harmonic moments are approximated by

$$\phi_{pq} = \sum_{m=1}^{M} Y_{pq}\left(\mu_m,\eta_m\right)\ \psi_m w_m,\quad \sigma_{pq} = \sum_{m=1}^{M} Y_{pq}\left(\mu_m,\eta_m\right)\ \sigma_m w_m, \qquad (9-d)$$

and the angular weights w_m are normalized such that $\sum_{m=1}^{M} w_m = 1.$

Solving the above set of quations constitute the *transport sweep*. Next we use the DSA to generate an estimate of the error in the transport fluxes. Final form of the acceleration equation is a diffusion like equation given by

$$A_{i+1,j+1}f^{(l+1)}_{00,i+3/2,j+3/2} + \left(C_{i+1,j} + C_{i+1,k+1}\right)f^{(l+1)}_{00,i+3/2,j+1/2}$$
$$+A_{i+1,j}f^{(l+1)}_{00,i+3/2,j-1/2} + \left(F_{i+1,j+1} + F_{ij+1}\right)f^{(l+1)}_{00,i+1/2,j+3/2}$$
$$\left(E_{ij} + E_{i+1,j} + E_{i,j+1} + E_{i+1,j+1}\right)f^{(l+1)}_{00,i+11/2,j+1/2}$$
$$+\left(F_{i+1,j} + F_{ij}\right)f^{(l+1)}_{00,i+1/2,j-1/2} + A_{i,j+1}f^{(l+1)}_{00,i-1/2,j+3/2}$$

$$+\left(C_{ij}+C_{i,k+1}\right)f_{00,i-1/2,j+1/2}^{(l+1)}+A_{ij}f_{00,i-1/2,j-1/2}^{(l+1)}$$

$$-\sum_{j'=j}^{j+1}\sum_{i'=i}^{i+1}g_{i'j'}^{(l+1/2)}\quad\sum_{j'=j}^{j+1}\left(\hat{\theta}_{i+1,j'}^{(l+1/2)}-\hat{R}_{ij'}^{(l+1/2)}\right)-\sum_{i'=i}^{i+1}\left(\tilde{\theta}_{i',j+1}^{(l+1/2)}-\tilde{R}_{i'j}^{(l+1/2)}\right). \qquad (10)$$

Solving the above equation iteratively produces the correction of the scaler flux. The correction to the other moments of the angular fluxes are calculated through the following equations:

$$f_{00,i\pm1/2,j}^{(l+1)}=\frac{f_{00,i\pm1/2,j+1/2}^{(l+1)}+f_{00,i\pm1/2,j-1/2}^{(l+1)}}{2}, \qquad (11-a)$$

$$f_{00,ij\pm1/2}^{(l+1)}=\frac{f_{00,i+1/2,j\pm1/2}^{(l+1)}+f_{00,i-1/2,j\pm1/2}^{(l+1)}}{2}. \qquad (11-b)$$

$$f_{10,ij'}^{(l+1/2)}=-\frac{1}{3h_i\left(\sigma_{s00,ij}-\sigma_{s10,ij}\right)}\left(f_{00,i+1/2,j}^{(l+1)}-f_{00,i-1/2,j}^{(l+1)}\right)$$

$$+\frac{\sigma_{s\ 10,ij}\left(\phi_{10,ij}^{(l+1/2)}-\phi_{10,ij}^{(l)}\right)}{\left(\sigma_{s00,ij}-\sigma_{s10,ij}\right)}, \qquad (11-c)$$

$$f_{11,ij'}^{(l+1/2)}=-\frac{1}{3k_j\left(\sigma_{s00,ij}-\sigma_{s11,ij}\right)}\left(f_{00,i,j+1/2}^{(l+1)}-f_{00,i,j-1/2}^{(l+1)}\right)$$

$$+\frac{\sigma_{s11,ij}\left(\phi_{11,ij}^{(l+1/2)}-\phi_{11,ij}^{(l)}\right)}{\left(\sigma_{s00,ij}-\sigma_{s11,ij}\right)}, \qquad (11-d)$$

$$\hat{f}_{00,ij}^{(l+1)}=\left[\gamma\left(2\hat{\rho}_{x,ij}\sigma_{ij}\sigma_{s,ij}k_j-2\rho_{x,ij}\sigma_{s,ij}\right)h_i\left(\hat{\phi}_{00,ij}^{(l+1/2)}-\hat{\phi}_{00,ij}^{(l)}\right)\right.$$

$$\left.+\left(3\hat{\rho}_{x,ij}\sigma_{ij}k_j-\frac{1}{2}\sigma_{ij}h_i+\gamma\rho_{x,ij}\sigma_{ij}h_i\right)\left(f_{00,i+1/2,j}^{(l+1)}-f_{00,i-1/2,j}^{(l+1)}\right)\right]$$

$$\times\left[6\hat{\rho}_{x,ij}\sigma_{ij}k_j+2\gamma h_i\left(\rho_{x,ij}\sigma_{s,ij}+\sigma_{r,ij}\hat{\rho}_{x,ij}\sigma_{ij}k_j\right)-\sigma_{ij}h\right]^{-1} \qquad (11-e)$$

$$\tilde{f}_{00,ij}^{(l+1)} = \left[\gamma\left(2\tilde{\rho}_{y,ij}\sigma_{ij}\sigma_{s,ij}h_i - 2\rho_{y,ij}\sigma_{s,ij}\right)k_j\left(\tilde{\phi}_{00,ij}^{(l+1/2)} - \tilde{\phi}_{00,ij}^{(l+1/2)}\right) \right.$$

$$\left. + \left(3\tilde{\rho}_{y,ij}\sigma_{ij}h_i - \frac{1}{2}\sigma_{ij}k_j + \gamma\rho_{y,ij}\sigma_{ij}k_j\right)\left(f_{00,i,j+1/2}^{(l+1)} - f_{00,i,j-1/2}^{(l+1)}\right)\right]$$

$$\times\left[6\tilde{\rho}_{y,ij}\sigma_{ij}h_i + 2\gamma k_j\left(\rho_{y,ij}\sigma_{s,ij} + \sigma_{r,ij}\tilde{\rho}_{y,ij}\sigma_{ij}h_i\right) - \sigma_{ij}k_j\right]^{-1} \qquad (11-f)$$

$$\tilde{f}_{10,ij}^{(l+1/2)} = \frac{\sigma_{s10,ij}\left(\phi_{10,ij}^{(l+1/2)} - \phi_{10,ij}^{(l)}\right)}{\left(\sigma_{s00,ij} - \sigma_{s10,ij}\right)}, \qquad (11-g)$$

$$\tilde{f}_{11,ij}^{(l+1/2)} = \frac{\sigma_{s11,ij}\left(\phi_{11,ij}^{(l+1/2)} - \phi_{11,ij}^{(l)}\right)}{\left(\sigma_{s00,ij} - \sigma_{s11,ij}\right)}, \qquad (11-h)$$

Finaly, the accelerated flux is computed from the correction terms by adding the correction to the corresponding flux calculated from the transport sweep. In case no acceleration was done, the correction terms are set to zero.

$$\phi_{pq,ij}^{(l+1)} = f_{pq,ij}^{(l+1)} + \phi_{pq,ij}^{(l+1/2)} \quad , \qquad (12-a)$$

$$\hat{\phi}_{pq,ij}^{(l+1)} = \hat{f}_{pq,ij}^{(l+1)} + \hat{\phi}_{pq,ij}^{(l+1/2)} \quad , \qquad (12-b)$$

$$\tilde{\phi}_{pq,ij}^{(l+1)} = \tilde{f}_{pq,ij}^{(l+1)} + \tilde{\phi}_{pq,ij}^{(l+1/2)} \quad , \qquad (12-c)$$

Outline of the TDMGDSA code

Figure 2 shows a schematic diagram of the TDMGDSA computer code. The user should first enter the problem description. To ease this step, another program GEN was written, which interactively generates the input file for TDMGDSA. Next, the various variables are initialized and computed. This includes the direction angles, weights, initial flux, and the various material dependent variables needed for the linear discontinuous and weighted diamond calculations. The code then enters the outer iteration loop. First the fission source is computed using the current flux values. Then for each energy group the appropriate fixed source problem cross sections and source are generated and fed into the fixed source calculation routine. The resulting flux is then stored back into the multigroup fluxes. This is repeated for all energy groups. Next, is the calculation of the error estimate of the group fluxes. If

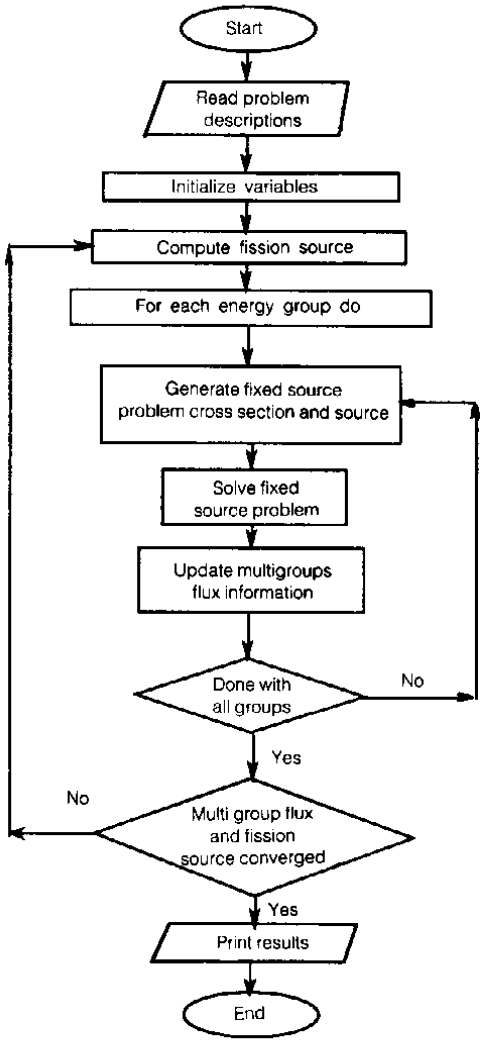

Fig. 2. Outline of the TDMGDSA code

this error and the fission source error is less then the error criterion then we have reached our goal, else we check first that we did not exceed the maximum number of outer iterations, if not a new iteration is performed. The last step, is to printout the results.

An outline of the fixed source calculations is shown in Fig. 3. Here first a transport sweep is performed for all cells for each direction [10]. If acceleration is on, the next step will be to compute the correction terms for the fluxes. An estimate of the error is computed next, after which we test against convergence. If the flux did not converge then we check that we did not exceed the maximum number of iterations, if not we perform a new iteration. Otherwise, the flux is returned back to the calling routine.

To save time, the maximum number of iteration in the fixed source calculation is reduced by a factor of ten until the fission source error becomes small. In this version this is done until the fission source error is less than the square root of the error criterion.

Acceleration Equations Solution Method

The low order acceleration equation can not be solved directly. An iterative method had to be used for solving these equations. In this code three numerical methods were implemented. Namely:

Gauss-Seidel iterative technique
Jacobi iterative technique with Chebychev acceleration [24]
Multigrid technique [20-23]

The first two are well known numerical techniques. Today, multigrid method is one of the hot topics in numerical science. In fact, multigrid algorithm is one of the reasons behind the success of the TWODANT code. In this project the black box multigrid routine [22] which utilizes a general form of the multigrid technique is used.

The idea behind the multigrid technique is that for certain iterative problems, the error in estimating the fluxes reduces as the mesh size increases. Therefore, by solving the problem at different mesh sizes, one can estimate the error at the fine grid from the error at the coarse grid. The problem is solved iteratively at all grids except the coarse mesh grid where it is solved exactly by inverting the matrix. The success of the method relies heavily on the good mapping of the flux from the fine to the coarse grid and the mapping of the error from the coarse to the fine grid.

Black box multigrid (BBOXMG) is a general multigrid method developed by Dendy [22] in which one needs only to specify the (logically rectangular, positive

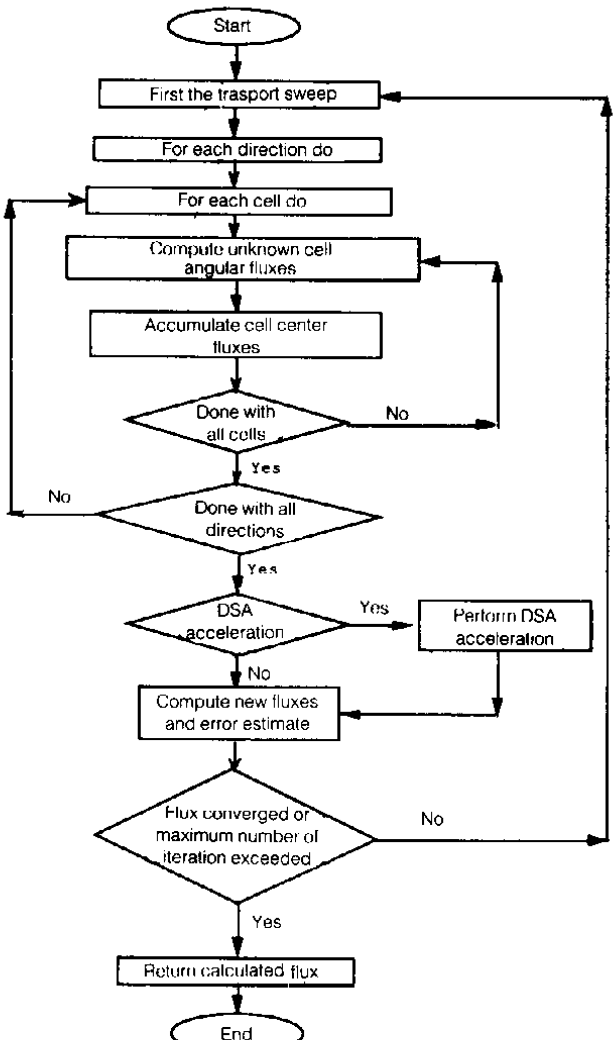

Fig. 3. An outline of the fixed source calculations.

definite) matrix problem and the BBOXMG does everything else necessary to set up the auxiliary coarser problems to achieve a multigrid solution. Black box multigrid, was designed to be used for general problems, thus making it suitable for casual quick usage. For production codes, a custom made multigrid algorithm has to be implemented. This is out of the scope of this project, but could be implemented in the future.

The code user can choose between the three different solution methods. In this project more emphasis was directed toward using the Jacobi iterative technique with Chebychev acceleration. Chebychev has the power of reducing the spectral radius considerably provided that it is less than unity. Currently, this is the preferred method, and it performs very well, usually faster than the general black box multigrid.

Results and Discussion

In this paper TDMGDSA a two dimensional multigroups diffusion synthetically accelerated neutron transport code is developed. The code uses the diffusion synthetic acceleration (DSA) method, to accelerate the transport calculations for diamond (DD), weighted diamond (WD), and linear discontinuous (LD) differencing schemes.

Different tests were performed to check the code against any coding or logical errors. All tests were run on a 486 IBM PC compatible personal computer. These tests consist of qualitative and quantitative tests. In the qualitative tests the code is checked against itself. By running the code on problems with similar material compositions but different orientation, the code could produce the same fluxes but now with different orientation. This is used to check against any coding errors. In the quantitative tests, the code is run on a problem, and the results are compared against either an analytic solution or the result of another computer code. The code was checked against results obtained from running ANISN [19] on an IBM RISC/6000 at King Saud University for a one dimensional slab problem.

The code was run on a homogeneous medium, one neutron energy group and with reflective boundary conditions to simulate an infinite homogeneous medium that can be solved analytically. The result agreed exactly with theory.

In the consistency tests, the problem is divided into four regions, each with different material composition as shown in Fig. 4. The code was run four times. In each time the materials were rotated. This was done to test the code against any logical errors. All four runs generated the same flux distribution.

The previous problem was rerun with reflective boundary conditions. Again the resulting flux agreed.

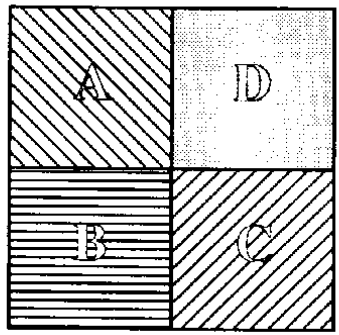

Fig. 4. Material distribution for test problem.

The code was tested against ANISN [19] a one dimensional multigroup transport code. Figure 5 shows the geometry of the problem, the cross sections are given in Tables 1 and 2. To simulate one dimensional problem, reflective boundary conditions where used with the top and bottom boundaries, diamond differencing was used. The two codes generated very similar fluxes. Figure 6 shows the computed flux.

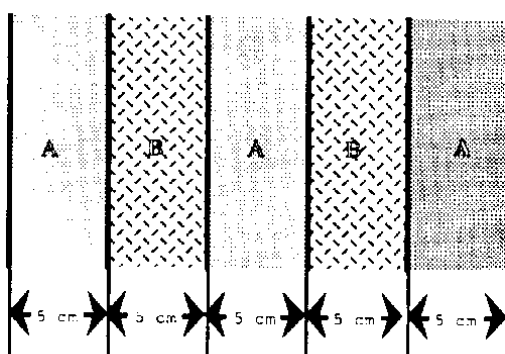

Fig. 5. Slab geometry problem test.

Table 1. Material A properties (Slab problem)

	σ_t	$V\sigma_f$	χ	$S_{00,g}$	$\sigma_{s00,g->1}$	$\sigma_{s00,g->2}$
Group 1	0.1	0.0	0.0	1.0	0.01	0.07
Group 2	2.2	0.0	0.0	0.0	0.0	2.1

Table 2. Material B properties (Slab problem)

	σ_t	$V\sigma_f$	χ	$S_{00,g}$	$\sigma_{S00,g->1}$	$\sigma_{S00,g->2}$
Group 1	0.1	0.04	1.0	1.0	0.07	0.0
Group 2	2.4	0.4	0.0	0.0	0.0	2.1

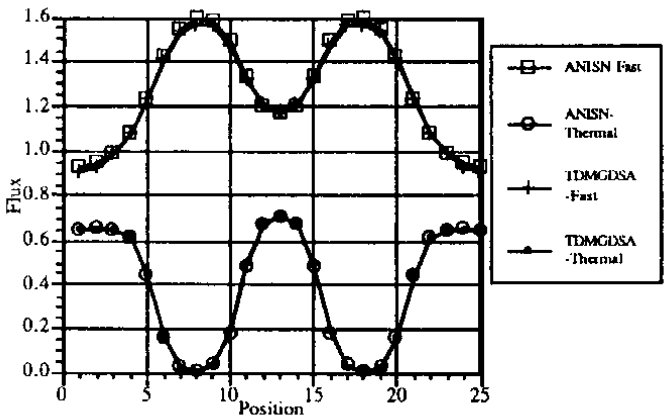

Fig. 6. Group fluxes in slab problem.

The previous four tests were to check the validity of the transport part of the code. In order to test the acceleration part, the code was applied to the problem shown in Fig. 7, with the material properties given in Tables 3 and 4. In this problem, an external source is uniformly distributed except at four locations where a fission source is present. In the first and second groups the self scattering cross section is small compared to the total cross section, while in the third group the problem is highly self scattering. This is similar to real life problems. This problem is solved both with and without acceleration. It was divided into 1 cm^2 square cells, S4 quadrature set was used. The flux was calculated for the DD, WD and LD differencing. In all cases the accelerated and unaccelerated flux calculations agreed, though, it took about four times as much time for the unaccelerated calculations to converge, as shown in Table 5.

Figures 8, 9 and 10 show the resulting flux. The peaks in the first two figures are located at material B which contain a fission source; these peaks are due to fission. In Fig. 10 the arrows point toward the flux valleys located at material B. This is caused by the self shielding effect produced by the fission source.

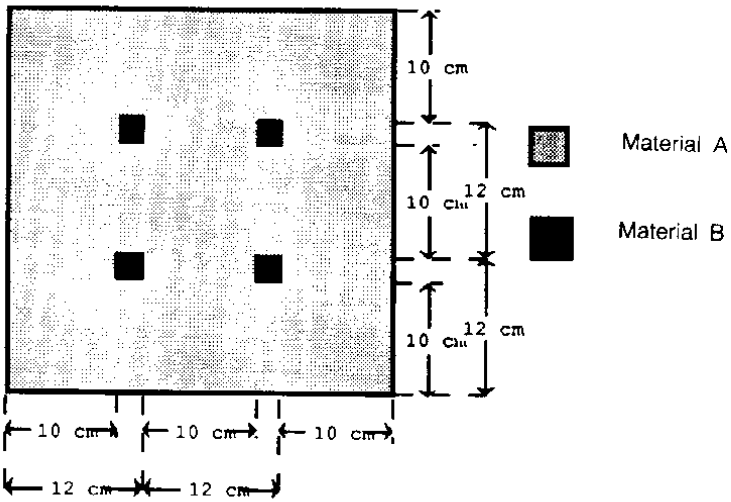

Fig. 7. Material distribution for three groups problem.

Table 3. Material A properties (2D problem)

	σ_t	$V\sigma_f$	χ	$\sigma_{s00,g->1}$	$\sigma_{s00,g->2}$	$\sigma_{s00,g->3}$	S_{00}
Group 1	0.3370	0.0	0.0	0.0	0.280	0.0150	1.0
Group 2	0.7	0.0	0.0	0.0	0.010	0.660	0.0
Group 3	2.1	0.0	0.0	0.0	0.0	2.080	0.0

Table 4. Material B properties (2D problem)

	σ_t	$V\sigma_f$	χ	$\sigma_{s00,g->1}$	$\sigma_{s00,g->2}$	$\sigma_{s00,g->3}$	S_{00}
Group 1	0.1245	0.0254	0.5750	0.0706	0.0327	0.018	0.0
Group 2	0.1558	0.1110	0.3260	0.0	0.1388	0.0107	0.0
Group 3	0.2372	0.0149	0.0990	0.0	0.0	0.2268	0.0

Table 5. Execution times for accelerated and unaccelerated test problem (2D problem)

	Transport	DSA	% gain
DD	111 min.	14 min.	793
WD	104 min.	16 min.	650
LD	781 min.	44 min.	1775

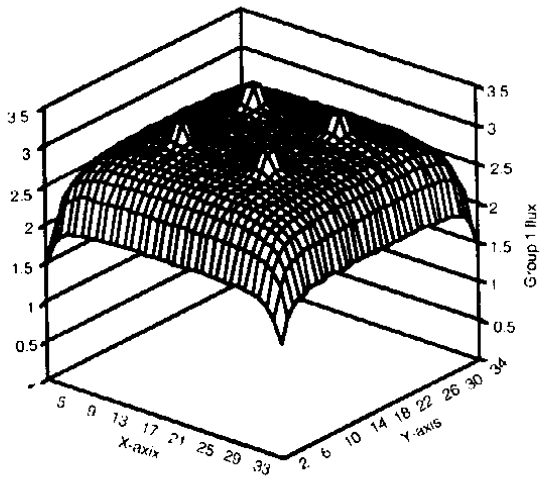

Fig. 8. Group one flux distribution

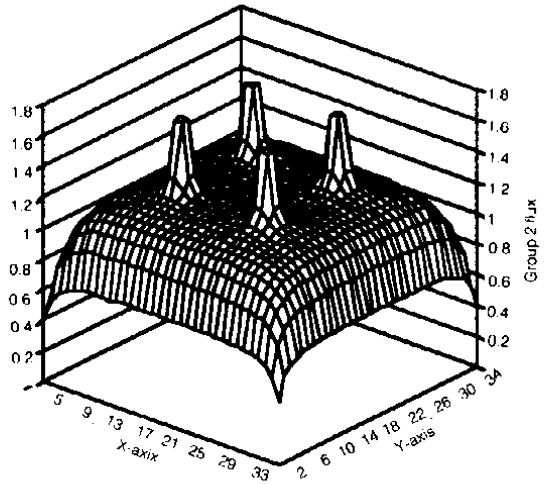

Fig. 9. Group two flux distribution

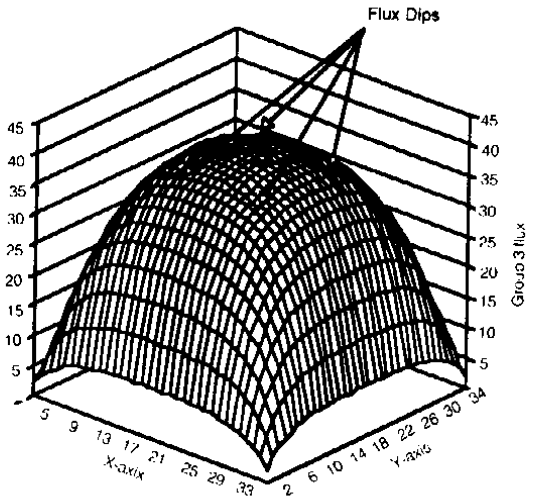

Fig. 10. Group three flux distribution

Conclusion and Recommendations

The TDMGDSA code, a multigroup discrete ordinates transport code in two dimensional X-Y geometry, implementing the diffusion synthetic acceleration for diamond, weighted diamond and linear discontinuous spatial differencing, was developed. Several tests were performed on the code and it is found that the code is free of design bugs, and that the diffusion synthetic acceleration does really work and save computation time. The lack of two-dimensional neutron transport codes some what limited the testing of TDMGDSA.

TDMGDSA fills an important need at our nuclear engineering department. It can be used for designing and studying various reactor and shielding problems. The diffusion synthetic acceleration becomes specially important because of the highly scattering nature of these problems. In addition, TDMGDSA is the first computer code that utilizes a linear and stable DSA implementation for the diamond differencing, and is one of the first codes to accelerate the weighted diamond and linear discontinuous transport calculations.

TDMGDSA was designed modularly. This allows the addition and modification of the underlying routines with ease. This is an important feature, which allows the code to be improved by different people.

TDMGDSA is in its infancy. Further improvements are required before it becomes a production code. In the following we will list some of them.

The black box multigrid method [22-23] (BBOXMG) used for solving the acceleration equations is a general form of the multigrid technique. The BBOXMG was designed for implementation for wide range of problems. This necessarily makes it slow, and inappropriate for production quality code. An improvement that can be introduced to the code is to design the multigrid method specifically for the diffusion synthetic equations.

More intelligence has to be put into the code. By this we mean educating the code to correctly decide which path to follow while solving the problem at hand. This includes when to quit iterating, what algorithm to use for solving the acceleration equation, and when to turn acceleration off. The latter is needed for problems with low scattering media at which DSA does not do any useful work and consume valuable CPU time.

Negative fluxes is a direct consequence of the spatial truncation error inherent in the spatial discretization. A negative flux treatment should be implemented.

Accurate flux calculation relies heavily on accurate cross-section calculations. The cross-sections are usually calculated using other codes designed specifically for that purpose. A requirement for this code to be useful is the availability of the cross-section generating codes along with the appropriate raw data. Currently non is available at the nuclear engineering department. These codes can be integrated into the TDMGDSA to allow one step run of the design or shielding problems.

It is our feeling that TDMGDSA can provide a well needed service for nuclear engineering in Saudi Arabia. This is one of many necessary tools that are needed to implement the nuclear technology in the Kingdom.

Acknowledgement. The authors are grateful to the Scientific Research Administration, College of Engineering, King Abdulaziz University and the Scientific Research Council, King Abdulaziz University, Jeddah, Saudi Arabia, for the research grant for this project.

Appendix A

$$h_i = \chi_{i+1/2} - \chi_{i-1/2}\ , \qquad k_j = y_{j+1/2} - y_{j-1/2}, \qquad v_{ij} = h_i k_j \tag{A-1}$$

$$\hat{\varepsilon}_{m,ij} = \frac{\sigma_{ij} h_i}{\mu_m}, \qquad \tilde{\varepsilon}_{m,ij} = \frac{\sigma_{ij} k_i}{\eta_m} \tag{A-2}$$

$$A_{ij} = -\frac{k_j}{h_i} D_{x,ij} - \frac{h_i}{k_j} D_{y,ij} + \frac{\nu_{ij}\sigma_{r,ij}}{4} \qquad (A-3)$$

$$C_{ij} = -\frac{k_j}{h_i} D_{x,ij} + \frac{h_i}{k_j} D_{y,ij} + \frac{\nu_{ij}\sigma_{r,ij}}{4} \qquad (A-4)$$

$$E_{ij} = \frac{k_j}{h_i} D_{x,ij} + \frac{h_i}{k_j} D_{y,ij} + \frac{\nu_{ij}\sigma_{r,ij}}{4} \qquad (A-5)$$

$$F_{ij} = \frac{k_j}{h_i} D_{x,ij} - \frac{h_i}{k_j} D_{y,ij} + \frac{\nu_{ij}\sigma_{r,ij}}{4} \qquad (A-6)$$

$$g_{00,ij}^{(l+1/2)} = \nu_{ij}\sigma_{s,ij}\left(\phi_{00,ij}^{(l+1/2)} - \phi_{00,ij}^{(l)}\right) \qquad (A-7)$$

$$\hat{g}_{ij}^{(l+1/2)} - \hat{\sigma}_{s,ij}\left(\hat{\phi}_{00,ij}^{(l+1/2)} - \hat{\phi}_{00,ij}^{(l)}\right) + 2k_j \frac{\sigma_{s10,ij}}{\sigma_{ij} - \sigma_{s10,ij}}\left(\phi_{10,ij}^{(l+1/2)} - \phi_{10,ij}^{(l)}\right) \qquad (A-8)$$

$$\tilde{g}_{ij}^{(l+1/2)} - \tilde{\sigma}_{s,ij}\left(\tilde{\phi}_{00,ij}^{(l+1/2)} - \tilde{\phi}_{00,ij}^{(l)}\right) + 2h_i \frac{\sigma_{s11,ij}}{\sigma_{ij} - \sigma_{s11,ij}}\left(\phi_{11,ij}^{(l+1/2)} - \phi_{11,ij}^{(l)}\right) \qquad (A-9)$$

$$D_{x,ij} = \frac{1}{3\left(\sigma_{ij} - \sigma_{s\ 10,ij}\right)} + B_{x,ij}, \qquad D_{y,ij} = \frac{1}{3\left(\sigma_{ij} - \sigma_{s\ 11,ij}\right)} + B_{y,ij} \qquad (A-10)$$

$$B_{x,ij} = \frac{h_i^2}{2} \frac{\Theta\left(\rho_{x,ij}\sigma_{s00,ij} + \sigma_{r00,ij}\hat{\rho}_{x,ij}\sigma_{ij}k_j\right) - \rho_{x,ij}\sigma_{ij}}{\Theta\left(6\hat{\rho}_{x,ij}\sigma_{ij}k_j + 2\gamma\ h_i\left(\rho_{y,ij}\sigma_{s00,ij} + \sigma_{r,ij}\hat{\rho}_{x,ij}k_j\right)\right) - \sigma_{ij}h_i}$$

$$B_{y,ij} = \frac{k_j^2}{2} \frac{\Theta\left(\rho_{y,ij}\sigma_{s,ij} + \sigma_{r,ij}\tilde{\rho}_{y,ij}\sigma_{ij}h_j\right) - \rho_{y,ij}\sigma_{ij}}{\Theta\left(6\tilde{\rho}_{y,ij}\sigma_{ij}h_j + 2\gamma\ k_j\left(\rho_{y,ij}\sigma_{s,ij} + \sigma_{r,ij}\tilde{\rho}_{y,ij}h_i\right)\right) - \sigma_{ij}k_j} \qquad (A-11)$$

$$\hat{\sigma}_{s,ij} = \Theta \frac{2\sigma_{s,00,ij}\left(\hat{\rho}_{x,ij}\sigma_{ij}k_j - \rho_{x,ij}\right)\nu_{ij}}{\left(6\tilde{\rho}_{x,ij}\sigma_{ij}k_j + 2\gamma\ h_i\left(\rho_{x,ij}\sigma_{s00,ij} + \sigma_{r,00ij}\hat{\rho}_{x,ij}k_j\right)\right) - \sigma_{ij}h_i} \qquad (A-12)$$

$$\tilde{\sigma}_{s,ij} = \Theta \frac{2\sigma_{s,ij}\left(\tilde{\rho}_{y,ij}\sigma_{ij}h_i - \rho_{y,ij}\right)v_{ij}}{6\tilde{\rho}_{y,ij}\sigma_{ij}h_i + 2\gamma\ \ k_j\left(\rho_{y,ij}\sigma_{s,ij} + \sigma_{r,ij}\tilde{\rho}_{y,ij}h_i\right) - \sigma_{ij}k_j} \qquad (A-13)$$

$$\rho_{x,ij} = \sum_{m=1}^{M} w_m \alpha_{m,ij}\mu_m, \rho_{y,ij} = \sum_{m=1}^{M} W_m \beta_{m,ij}\eta_m \qquad (A-14)$$

$$\theta_x^{(l+1/2)} = \sum_{m=1}^{M} w_m \hat{\Theta}_m \psi_m^{(l+1/2)} - \gamma\phi_{10}^{(l+1/2)} \qquad (A-15)$$

$$\theta_y^{(l+1/2)} = \sum_{m=1}^{M} w_m \hat{\Theta}_m \psi_m^{(l+1/2)} - \gamma\phi_{11}^{(l+1/2)} \qquad (A-16)$$

$$\gamma = 3 \sum_{\mu_m > 0} w_m \mu_m \qquad (A-17)$$

$$\hat{\rho}_{x,ij} = \sum_{m=1}^{M} \frac{3\alpha_{m,ij}\mu_m\eta_m^2}{\sigma_{ij}k_j} w_m \ , \quad \tilde{\rho}_{y,ij} = \sum_{m=1}^{M} \frac{3\beta_{m,ij}\mu_m^2\eta_m}{\sigma_{ij}h_i} w_m \qquad (A-18)$$

References

[1] Hill, T. "ONETRAN: A Discrete Ordinates Finite Element Code for the Solution of One Dimensional Multigroup Transport Equation." *Los Alamos Scientific Laboratory*, LA-5990-MS (1975).

[2] Kopp, H. "Synthetic Method Solution of the Transport Equation." *Nucl. Sci. Eng.*, 17, (1963), 65.

[3] Crawford, B. and Friedman, J. "A Two-Dimensional Application of an Iterative Method for Solving the Neutron Transport Equation with Anisotropic Scattering." *Trans. Am Nucl. Soc.*, 9 (1966), 472.

[4] Gelbard, E. and Hageman, L. "The Synthetic Method as Applied to the S_n Equations." *Nucl. Sci. Eng.*, 32 (1969), 288.

[5] Reed, W. "The Effectiveness of Acceleration Techniques for Iterative Methods in Transport Theory." *Nucl. Sci. Eng.*, 45 (1971), 237.

[6] Alcouffe, R. "A Stable Diffusion Synthetic Acceleration Method for Neutron Transport Iterations." *Trans. Am. Soc.*, 23 (1976), 203.

[7] Alcouffe, R. "The Diffusion Synthetic Acceleration Method Applied to Two-Dimensional Neutron Transport Problems." *Trans. Am. Soc.*, 27 (1977), 347.

[8] Alcouffe, R. "Diffusion Synthetic Acceleration Methods for the Diamond-Differenced Discrete-Ordinates Equations." *Nucl. Sci. Eng.*, 64 (1977), 344.

[9] Larsen, E. "Unconditionally Stable Diffusion-Synthetic Acceleration Methods for the Slab Geometry Discrete-Ordinates Equations. Part I: Theory." *Nucl. Sci. Eng.*, 82 (1982), 47.

[10] Lewis, E. and Miller, W. *Computational Methods of Neutron Transport*. New York: John Wiley & Sons, 1984.

[11] Larsen, E. "Diffusion-Synthetic Acceleration Methods for Discrete-Ordinates Problems." *Trans. Theory and Stat. Phys.*, 13 (1984), 107.

[12] Alcouffe, R., Brinkley, F., Marr, D. and O'Dell, R. "User's Guide for TWODANT: A Code Package for Two-Dimensional, Diffusion-Accelerated, Neutron-Particle Transport." *Los Alamos National Laboratory*, LA-10049-M, (1984).

[13] AboAlfaraj, T. *Diffusion Synthetic Acceleration of the Diamond Differenced and Linear Discontinuous Discrete Ordinates Equations in X-Y Geometry*. Ph.D. Thesis, University of Michigan, 1989.

[14] AboAlfaraj, T. and Larsen, E. "The Efficiency of Linear and Nonlinear Implementations of Diffusion Synthetic Acceleration." *Proceedings of Advances in Mathematics, Computations and Reactor Physics*, 3 (1991), 11.1 2-1.

[15] AboAlfaraj, T. and Melaibari, A. *Development of a Multigroup Two Dimensional Synthetic Acceleration Neutron Transport Computer Code in X-Y Geometry*. Project No. 059/410, Jeddah: King Abdulaziz University, 1990.

[16] Carleson, B. and Lathrop, K. *Transport Theory – The Method of Discrete Ordinates in Computing Methods in Reactor Physics*. New York: Gordon and Breach, 1968.

[17] Larsen, E. and O'Dell, R. "The Diamond Difference Method on Optically Thick Spatial Meshes." *Los Alamos Scientific Laboratory*, LA-11589-PR, (1989).

[18] Larsen, E. and O'Dell, R. "Using ONEDANT and TWODANT to Generate Diamond-Differenced S_N Solutions Without Negative Flux Fixup." *Los Alamos Scientific Laboratory*, LA-11589-PR (1989).

[19] Engle, W. *The Users Manual for ANISN: A One-Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering*. Oak Ridge Gaseous Diffusion Plant, 1967.

[20] Briggs, W. *Multigrid Tutorial*. Pennsylvania: Society of Industrial and Applied Mathematics, 1987.

[21] Mc Cormick S. (editor) *Multigrid Methods*. Pennsylvania: Society of Industrial and Applied Mathematics, 1987.

[22] Dendy, J. *Black Box Multigrid*. Los Alamos Scientific Laboratory, LA-UR-81-2337.

[23] Dendy, J. "Black Box Multigrid." *J. Comp. Phys.* 48 (1982), 366.

[24] Hageman, L. and Young, D. *Applied Iterative Methods*. New York: Academic Press, 1981.

برنامج حساب التوزيع الانتقالي للنيوترونات المتعددة الطاقة في بعدين باستخدام التسريع الانتشاري التوليفي (TDMGDSA)

طارق ج. أبو الفرج وعبدالغني م. ميلاياري

قسم الهندسة النووية، جامعة الملك عبدالعزيز، جدة، المملكة العربية السعودية

(استُلم في ١٩٩١/١٢/٣م؛ قُبل للنشر في ١٩٩٢/٥/١٨م)

ملخص البحث. تعتبر برامج حساب التوزيع الانتقالي للنيوترونات أدوات أساسية في تصميم المفاعلات النووية والدروع الإشعاعية. ويقتضي تطبيق هذه الحسابات تكرار استخدام طرق حسابية معينة، ويزيد هذا التكرار كلما نقصت نسبة احتمال امتصاص النيوترونات إلى احتمال اصطدامها، مما استوجب تطوير وسائل تسريع لهذه الحسابات. ويعتبر التسريع الانتشاري التوليفي أفضلها على الإطلاق.

في هذا البحث نعرض تطوير برنامج لحساب التوزيع الانتقالي للنيوترونات المتعددة الطاقة في بعدين باستخدام التسريع الانتشاري التوليفي (TDMGDSA). ويتم في هذا البرنامج حل معادلة الزوايا المحددة للنيوترونات في بعدي س ـ ص وباستخدام طريقة التسريع الانتشاري التوليفي لثلاث طرق تقسيمية في الأبعاد هي التقسيم الماسي (Diamond Differencing) والتقسيم الماسي الموزون (Weighted Diamond Differencing) والتقسيم الخطي غير المتصل (Linear Discontinuous Differencing).

وقد تم إجراء عدة اختبارات على البرنامج، وقد وجد أنه سليم من الأخطاء ويعمل بفعالية.