CONF-910414--39

DE93 003463

~~APPENDIX 1~~

Proc. Int. Mtg. Advances in Mathematics

Computation and Reactor Physics

Pittsburgh, April, 1991

A Discrete Ordinate Response Matrix Method for Massively Parallel Computers

U.R. Hanebutte and E.E. Lewis
Department of Mechanical Engineering
Northwestern University
Evanston IL, 60208

FG02-88ER12810

Abstract

A discrete ordinate response matrix method is formulated for the solution of neutron transport problems on massively parallel computers. The response matrix formulation eliminates iteration on the scattering source. The nodal matrices which result from the diamond-differenced equations are utilized in a factored form which minimizes memory requirements and significantly reduces the required number of arithmetic operations per node. The red/black solution algorithm utilizes massive parallelism by assigning each spatial node to a processor. The algorithm is accelerated effectively by a synthetic method in which the low-order diffusion equations are also solved by massively parallel red/black iterations. The method has been implemented on a 16k Connection Machine-2, and S_8 and S_{16} solutions have been obtained for fixed-source benchmark problems in X-Y geometry.

I. Introduction

For two decades discrete ordinate methods[1,2] have frequently been the method of choice for the solution of a wide variety of deterministic transport problems. The combination of iteration on the scattering source, accelerated by synthetic or other methods, with algorithms for marching across the space-angle grid in the directions of neutron travel has been ideally suited for serial computers with nominal amounts of memory. In addition to providing effective computational solutions on serial computers, discrete ordinate methods often could be applied to large problems for which competing spherical harmonics, finite element or collision probability methods exceeded the available memory.

In the past few years massively parallel computers of the single instruction multiple data (SIMD) type[3,4] have emerged as exceedingly powerful tools for scientific computation. However, the thousands of parallel processors, each with local memory, present a radically different architecture from the serial machines for which discrete ordinate methods were originally developed. In computational continuum mechanics some of the most successful applications of massively parallel machines have come about through the reformulation of finite element and finite difference algorithms to associate one or more processors with each spatial mesh point.[5-7] Therefore, it seems likely that similar approaches may prove effective in the use of massively parallel SIMD machines for the solution of the discrete ordinate equations.

In this study we formulate the discrete ordinate response matrix method and develop a massively parallel algorithm for the solution of two-dimensional X-Y problems on a 16k processor Connection Machine-2[4]. The method has three salient features. First, the response matrix formulation utilizes a red-black algorithm in which processors operate simultaneously on approximately one half of the spatial cells and pass data only between nearest neighbors. Second, as in earlier work in radiation transport,[8,9] the unique structure the discrete ordinate equations is utilized to reduce greatly both the number of response matrix eleme which must be stored in local memory and the number of floating point operations associated with e nodal response matrix calculation. Third, synthetic acceleration[2,10,11] is adapted for use in conjunction w discrete ordinate response matrices, and a massively parallel algorithm for the solution of the associa five-point difference approximation to the diffusion equation is developed.

II. Response Matrix Algorithm

With conventional S_N quadratures,[1,11] discrete ordinate response matrix equations may be construct in X-Y geometry by gathering the $N(N+2)/4$ incoming and outgoing angular flux directions on each of four cell interfaces into two vectors Ψ^- and Ψ^+, each of which has dimension $M = N(N+2)$. The la number of directions leads to a response matrix, R , with a large dimension. Thus, if the cellular respo matrix operations,

$$\Psi^+ = \mathbf{R}\,\Psi^- + \mathbf{b}\,S\,,$$

are carried out without taking advantage of the structure of the equations, large memory requireme and numbers of floating point operations per node are encountered. For example, in an S_8 calculation response matrix has a dimension of 80 and therefore contains 6,400 elements.

Fortunately, both the memory requirements and the number of arithmetic operations per node can greatly reduced by utilizing the structure of the discrete ordinate equations. For clarity we introduce vec notation to outline the procedure; detailed expressions are given in Appendix A for the diamond differe equations. To begin, we combine the balance equation with the diamond difference relations to obtain scalar flux ϕ at the cell center in terms of the incoming angular flux distribution and the scattering sour

$$\phi = \mathbf{v}^T\,\Psi^- + \alpha\,Q\,,$$

where in conventional notation the scattering source is given by

$$Q = \sigma_s\,\phi + S\,.$$

Likewise, the outgoing flux distribution from the cell may be written as

$$\Psi^+ = \mathbf{H}\,\Psi^- + \mathbf{u}Q$$

In these equations the elements of the vectors u and v and of the matrix H are given in terms of the dimensions, the angular weights and direction cosines of the discrete ordinates. The critical property of matrix H is that it has only two nonzero elements per row or a total of $2M$ nonzero elements.

From the foregoing equations we may construct response matrix algorithms which obviate iteration the scattering source. By eliminating Q we obtain

$$\phi = (1-\alpha\,\sigma_s)^{-1}\,\mathbf{v}^T\,\Psi^- + (1-\alpha\,\sigma_s)^{-1}\,\alpha\,S$$

and

$$\Psi^+ = \mathbf{H}\,\Psi^- + \mathbf{u}(\sigma_s\,\phi + S)\,.$$

The two step procedure then consists of calculating first the scalar flux at the cell center and then outgoing flux distribution. This allows discrete ordinate response matrix calculations to be performed w only $5.5M$ floating point operations per cell as opposed to the $\approx 2M^2$ that would be required for solving linear algebra problem using the dense and nonsymmetric discrete ordinate response matrix. Likewise, o $1.25M$ memory locations are required for the response matrix as opposed to the $\approx M^2$ needed to store matrix in conventional form. That the two approaches are formally equivalent is easily seen by combin equation (5) and (6) to yield the form of Eq.(1).

$$\Psi^+ = \left[\mathbf{H} + (1-\alpha\,\sigma_s)^{-1}\sigma_s\mathbf{u}\mathbf{v}^T\right]\Psi^- + \left[1 + (1-\alpha\,\sigma_s)^{-1}\alpha\sigma_s\right]\mathbf{u}\,S\,.$$

To proceed from the foregoing equations, which pertain to a particular node, to a solution algorithm, we partition the problem domain into a checkerboard of red and black nodes in which the Ψ^+ emanating from the red nodes constitute the Ψ^- entering the black, and conversely. This two-cyclic red-black solution algorithm for the response matrix equations efficiently utilizes both the data-parallel structure and the nearest neighbor transfer capabilities of the Connection Machine-2. The convergence of the algorithm, however, typically is slow, characteristic of response matrix algorithms when they are applied to meshes fine enough for diamond differencing to be a reasonable approximation. The availability of effective acceleration methods therefore becomes an important issue.

We have examined both Chebyshev[12,13] and synthetic[10,11] acceleration methods for use in conjunction with the foregoing algorithm. We first employed a Chebyshev algorithm[13] for nonsymmetric red/black partitioned matrices with small complex eigenvalues. The method was not successful, we suspect, because the two-cyclic discrete ordinate iteration matrix contains eigenvalues with imaginary parts that are too large. The synthetic method described in the following section has been successful in accelerating the discrete ordinate response matrix calculation.

III. Synthetic Acceleration

Synthetic acceleration methods utilize a low-order operator, often from diffusion theory, to accelerate the higher-order transport operator. To formulate a synthetic acceleration method for use in conjunction with the S_N response matrix equations we begin with the within-group transport equation in the operator form[11]

$$H\,\psi = S\,,$$

where

$$H\,\psi = \hat{\Omega}\,\vec{\nabla}\,\psi + \sigma\,\psi - \sigma_s \int d\Omega\,\psi\,, \tag{7}$$

and $\psi(\vec{r},\hat{\Omega})$ is the angular flux. The diffusion equation is employed as the low-order operator, H_L. Thus we define

$$H_L\,\phi = -\vec{\nabla}\,D\,\vec{\nabla}\,\phi + \sigma_a\,\phi \tag{8}$$

where $\sigma_a = \sigma - \sigma_s$, $D = 1/(3\sigma)$, and $\phi(\vec{r})$ is the scalar flux. To obtain the synthetic acceleration equations the low-order operator and the transport operator are combined to yield

$$[H_L + H - H_L]\,\psi = S$$

and

$$H_L\,\psi = S - [H - H_L]\psi\,.$$

This equation may now be used to express the accelerated ψ^{l+1}, the $(l+1)^{\text{th}}$ iterate, in terms of $\tilde{\psi}^l$, the value obtained in the absence of acceleration,

$$H_L\,\psi^{l+1} = S - [H - H_L]\,\tilde{\psi}^l\,,$$

or correspondingly :

$$H_L\,(\psi^{l+1} - \tilde{\psi}^l) = S - H\,\tilde{\psi}^l\,. \tag{9}$$

We combine Eqs.(7) and (8) with Eq.(9) and then integrate over Ω as well as the volume V of one node. Utilizing Green's theorem, we obtain for each node

$$\left[-\int d\Gamma\,\hat{n}\cdot D\,\vec{\nabla} + \sigma_a \int dV\right](\phi^{l+1} - \tilde{\phi}^l) = \int dV\,[S - \sigma_a\tilde{\phi}^l] - \int d\Gamma \int d\Omega\;\hat{n}\cdot\hat{\Omega}\,\tilde{\psi}^l\,, \tag{10}$$

where Γ denotes the node surface, and integration over angle is normalized by $\int d\Omega = 1$. The quantity on the right hand side is recognized as the residual of the nodal neutron balance equation and will thus vanish as $\tilde{\psi}^l$ converges to the true solution.

To reduce Eq.(10) to a form compatible with the discrete ordinate response matrix formulation, the source and scalar flux values on the right are replaced by the cell-centered values ϕ and S of Eqs (2) and (3). The angular integral is approximated by the S_N quadrature, thus allowing the surface integral to be evaluated as a superposition of the node interface values of Ψ^+ and Ψ^-, the discretized angular flux appearing in Eqs.(2) through (6). Finally, the integrated diffusion operator on the left hand side is the point of departure from which the standard five-point cell-centered difference approximation may be obtained.[1]

The accelerated iteration scheme proceeds as follows. Let $\tilde{\Phi}^l$ be the global vector consisting the values of $\tilde{\phi}^l$ at each of the node centers. After a number of red/black discrete ordinate iterations the residual vector r can be constructed from the right side of Eq.(10). The global form of Eq.(10) then appears as

$$\mathbf{M}\,(\Phi^{l+1} - \tilde{\Phi}^l) = \mathbf{r}\,,$$

where M is the five-point difference operator. Finally, the accelerated iterate for the global flux vector is obtained from

$$\Phi^{l+1} = \tilde{\Phi}^l + \mathbf{M}^{-1}\,\mathbf{r} \tag{11}$$

and used to update the nodal scalar flux ϕ in Eqs.(3) and (6).

The solution indicated formally in Eqs.(11) is carried out by applying red-black iterations to the five-point diffusion equations. More precisely, an initial flux guess is obtained by solving the cell-centered diffusion equation. This solution is accelerated by using Chebyshev acceleration. The spectral radius and therefore the optimum over-relaxation factor for the diffusion operator are obtained as byproducts of these calculations. Thus, in successive applications of the diffusion operator for synthetic acceleration, relatively few optimum over-relaxed red/black iterations are needed to obtain sufficiently accurate values of Φ^{l+1} from Eq.(11).

IV. Results

The discrete ordinate response matrix method with synthetic acceleration has been implemented in a form suitable for massively parallel SIMD machines.[3,4] Red/black algorithms in which one processor is assigned to each spatial node are employed both for the discrete ordinate response matrix iterations and for the solution of the finite-differenced diffusion equations used in synthetic acceleration. The calculations which follows are carried out on a Connection Machine-2[4] (CM2) with 16,384 (2^{14}) processors, 8 kilobytes of local memory per processor and a 32 bit floating-point acceleration chip associated with each 32 processors. The coding is carried out on the Sun work station which also serves as a control processor to broadcast instructions to the Connection Machine. The code is written in CM-Fortran,[16] an implementation of Fortran 77 supplemented with array- processing extensions.

On fine-grained parallel machines both the computational efficiency and the size of the problems that can be accommodated are sensitive to the data structures which are employed. On the CM2 computation times per iteration are independent of the number of physical processors used and therefore of the number of spatial nodes in the red/black algorithms. Moreover, one may use either an 8k partition or the full set of processors. In addition, all array dimensions must be powers of two. Spatial grids of arbitrary size are treated by increasing the number of grid points to the next higher value of 2^n and masking operations on the unneeded processors.

The CM2 also allows 2^n virtual processors to be associated with each physical processor. This streamlines coding by increasing the apparent number of available processors. Substantially higher computing rates than those with explicit serial coding are obtained in situations where multiple-dimension arrays require more than the number of available physical processors. We have increased the computing rates (MFLOP'S) in red/black algorithms by as much as a factor of four by going from serial coding to the use of virtual processors. The gain are most dramatic in single precision where use is made of the floating-point accelerator chips.

In the discrete ordinate algorithms each of the $N(N+2)/8$ neutron direction in a quadrant is assigned a virtual processor. Then, as indicated in Appendix B, manipulations are performed on the sixteen angular flux vectors corresponding to the four quadrants on each of the four node interfaces. In addition to increasing the rate of arithmetic operations, this vector structure facilitates the fast transfer of flux data between

nearest neighbor processors which correspond to adjacent spatial nodes The 2^n restriction on vector length, however, does mean that some of the virtual processors are masked; for example in S_8 calculations only ten of sixteen are used. The 8 kilobytes of local memory per physical processor must be divided between virtual processors and local code storage requirements. In the discrete ordinate algorithms this results in a limitation of 32 virtual processors per physical processor. Thus in S_{16} calculations, for example, we must assign two physical processors per spatial node to accommodate the 36 directions per quadrant.

To examine the behavior of the data-parallel algorithms for the solution of the discrete ordinate response matrix equations a benchmark fixed-source muitiregion problem[17] has been utilized. In Fig. 1 the three solutions obtained on the CM2 with a 60x60 spatial grid are compared far from the source region. As expected, the S_8 and S_{16} are in good agreement aside from the small ray effects. The diffusion results come from the red-black solution of the cell-centered five-point difference equations also used for synthetic acceleration In Fig. 2 is shown a topological plot of $\log \phi$ for the S_8 solution on a 120x120 grid.

For the S_8 approximation times of the order of .15 Sec. per red/black iteration are obtained for spatial grids less or equal to 128x128. With these large numbers of grid points the single precision (32bit) response matrix algorithm exceeds 45 MFLOPS on the 16k CM2 with a virtual processor ratio of 16. A large S_8 problem with 128x256 grid points was treated by allocating two nodes and 32 virtual processors per physical processor Here computing rates exceed 60 MFLOPS.

Typically, mesh spaces of the order of a mean free path or less are employed in discrete ordinate calculations As a result, the convergence of the unaccelerated red/black response matrix algorithm is quite slow, often requiring hundreds of iterations. The dramatic improvements obtained from employing the synthetic acceleration method presented in Section III are illustrated in Fig 3. Here the error norm[15]

$$\delta^{l+1} = \left\| \Phi_B^{l+1} - \Phi_B^l \right\|_2 \Big/ \left\langle \Phi_B^{l+1} , \Phi_B^l \right\rangle^{1/2} ,$$

where Φ_B^l is the global vector of black node scalar flux values after l iterations, is plotted vs. the number of iterations.

The two broken lines in Fig. 3 indicate the behavior of the unaccelerated iteration with initial guess of zero flux and of a diffusion theory solution, respectively. As expected, the slopes of the lines, and therefore the asymptotic convergence rates, are identical. The slopes of the solid lines indicate the large improvements in convergence rates which are obtained when synthetic acceleration is employed at successively shorter intervals. When synthetic acceleration is employed after each four iterations, the average time per iteration increases by approximately 40%. To limit the times expended on diffusion calculations optimal over-relaxation factors are obtained as a byproduct of the diffusion solution used as an initial flux guess. As a result, a sufficiently accurate solution of the low-order diffusion operator can thereafter be obtained with relatively few red/black diffusion iterations. As indicated in Table I the use of synthetic acceleration results in significant decreases in total computing times.

V. Discussion

The results presented in the preceding section lend encouragement to the further pursuit of response matrix methods on massively parallel computers. Existing Connection Machines with larger numbers of processors and expanded local memory per processor[3] should allow the foregoing discrete ordinate formulation to be applied to three-dimensional problems of reasonable size. The inclusion of anisotropic scattering and the ability to treat multigroup and criticality problems also seems feasible with the availability of more powerful machines The red/black approach should also allow response matrices incorporating other space-angle approximations to be treated effectively. We are currently examining variational nodal response matrices which incorporate spherical harmonics approximations,[18,19] and we anticipate that nodal discrete ordinate methods may also be tractable on data-parallel machines.

Acknowledgements

All Connection Machine-2 calculations were performed at the Advanced Computing Research Fac Mathematics and Computer Science Division, Argonne National Laboratory. This work was supporte part by Department of Energy Grant DE-FG07-88ER12810.

References

[1] B.C. Carlson and K.D. Lathrop, "Transport Theory - The Method of Discrete Ordinat *Computing Methods in Reactor Physics*,H. Greenspan, C.N. Kelber ans D. Okrent (eds.) Gordon Breach, NY, 1968 Chapt 3

[2] R.E. Alcouffe, *Nucl. Sci. Eng. 64*, 344 (1977)

[3] P.J. Denning and W.F. Tichy , *Science, 250*, 1217 (1990)

[4] Connection Machine Model CM-2 Technical Summary, Thinking Machines Corp. T Report HA87-4, April 1987

[5] H.S. Simon (Ed), *Scientific Applications of the Connection Machine* , World Scientific Pub. neck, NJ 1989

[6] O.A. McBryan, *Parallel Comp. 9*, 1 (1988/89)

[7] P. Olsson and S.L. Johnsson, *Parallel Comp. 14*, 1 (1990)

[8] W.F. Miller Jr. and E.E. Lewis, "Nonlinear Response Matrix Formulation for the Solut of Time-Dependent Radiative Transfer Problems," *Trans. Int. Topl. Meeting on Adv. in Rea Physics, Mathematics and Computation*, Paris, April 27-30, 1986

[9] E.E. Lewis, U.R. Hanebutte and R.F. Stefan, *Trans. Th. Statist. Phys., 19* , 3-5, 387 (19

[10] H.J. Kopp, *Nucl. Sci. Eng. 17*, 65 (1963)

[11] E.E. Lewis, W.F. Miller Jr., *Computational Methods of Neutron Transport* , Wiley, New Y 1984

[12] R.S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962

[13] L.H. Hageman, D.M. Young, *Applied Iterative Methods*, Academic Press, New York, 1981

[14] G.I. Bell and S. Glasstone, *Nuclear Reactor Theory*, Van Nostrand Reinhold, NY, 1970

[15] K.L. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite Differenc Diffusion Theory Problems," ANL-8264, Argonne National Laboratory (1984)

[16] CM Fortran Reference Manual, Thinking Machines Corp. Version 5.1A-0.5, May 1989

[17] H. Khalil, *Nucl. Sci. Eng. 90*, 263 (1985)

[18] E.E. Lewis, *Nucl. Sci. Eng. 102*, 140 (1989)

[19] C.B. Carrico and E.E. Lewis, "Variational Nodal Solution Algorithms for Multigroup Criticality Problems", to appear in *Proc. Int. Topl. Meeting Advances in Mathematics, Computations and Reactor Physics*, April 28 - May 1, 1991 Pittsburgh, Pa

Table I
Timing Comparison for a S_8 60 x 60 Problem on 8k Processors

	Unaccelerated		Accelerated	
	$N = \infty$	$N = \infty$	$N = 20$	$N = 4$
time/iteration CPU [sec]	0.1446*	0.2053	0.2165	0.2805
number of iterations	476	476	222	84
total time CPU [sec]	68.8*	97.7	54.9	30.4

* Single Precision Computation
$N \equiv$ Iterations between acceleration step.

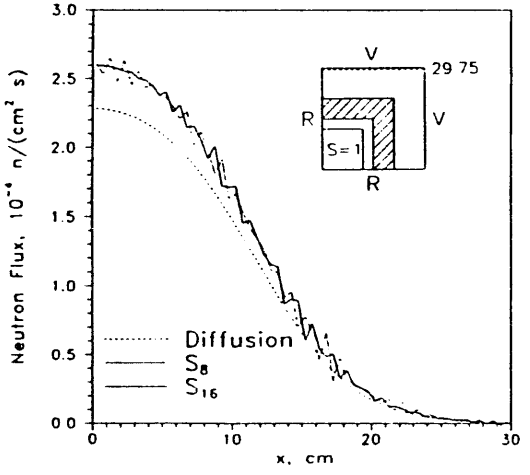

Fig. 1. Neutron Flux at y = 29.75 cm for a Multi Region Problem[17].

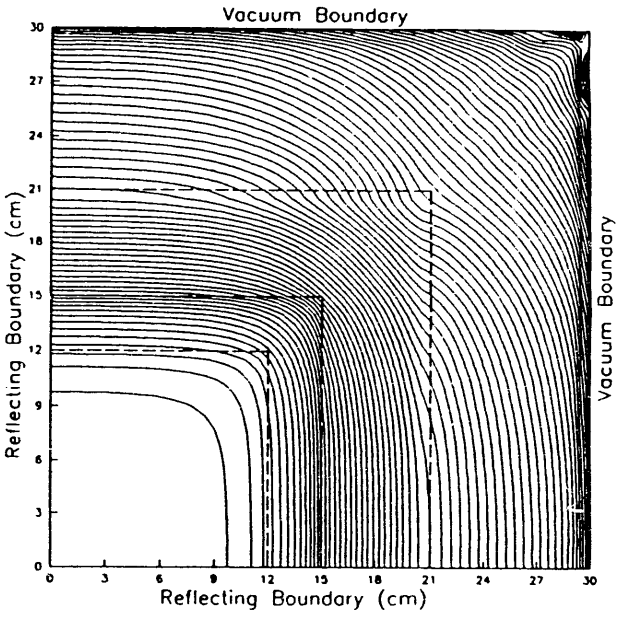

Fig. 2. Logarithmic Flux Plot Obtained from a 120x120 Mesh S_8 Response Matrix Algorithm

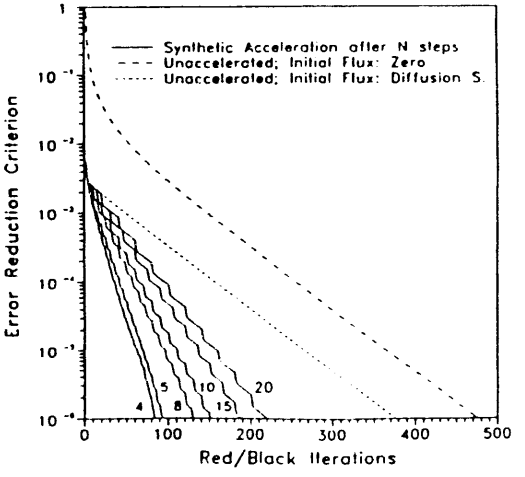

Fig. 3. Error Criterion vs. Iteration Number for Accelerated and Unaccelerated S_8 Calculations

Appendix A

The detailed expressions for the vectors and matrices appearing in Eqs.(1) through (5) may be derived from the discrete ordinate balance equation and the associated diamond difference relationships. In conventional notation[11] the balance equation for X-Y geometry is

$$\mu_n \frac{(\psi_{n,i+1/2,j} - \psi_{n,i-1/2,j})}{\Delta x_i} + \eta_n \frac{(\psi_{n,i,j+1/2} - \psi_{n,i,j-1/2})}{\Delta y_j} + \sigma_{i,j}\ \psi_{n,i,j} = \sigma_{s\,i,j}\ \phi_{i,j} + S_{i,j}\ ,$$

where n refers to angular direction $\hat{\Omega}_n$, with direction cosines μ_n and η_n, and the cell centered at x_i , y_j has interfaces at $x_{i\pm1/2}$ and $y_{j\pm1/2}$. The scalar flux at the cell center is represented as the quadrature

$$\phi_{i,j} = \frac{1}{4} \sum_{n=1}^{N(N+2)/2} w_n\ \psi_{n,i,j}\ ,$$

with weights w_n, and the diamond difference relations are

$$\psi_{n,i,j} = \frac{1}{2}\left(\psi_{n,i+1/2,j} + \psi_{n,i-1/2,j}\right),$$
$$\psi_{n,i,j} = \frac{1}{2}\left(\psi_{n,i,j+1/2} + \psi_{n,i,j-1/2}\right).$$

The scalar flux equation corresponding to Eq.(4) is obtained by using the diamond difference relationship to eliminate the outgoing angular flux from the balance equation, and then integrating $\psi_{n,i,j}$ over angle. We obtain

$$\phi_{i,j}^{l+1} = (1-\alpha\sigma_{s\,i,j})^{-1}\Bigg[\alpha S_{i,j} + \frac{1}{4}\sum_{\mu_n>0} w_n\tilde{\mu}_n\psi^l_{n,i-1/2,j} + \frac{1}{4}\sum_{\mu_n<0} w_n\tilde{\mu}_n\psi^l_{n,i+1/2,j} + \frac{1}{4}\sum_{\eta_n>0} w_n\tilde{\eta}_n\psi^l_{n,i,j-1/2} + \frac{1}{4}\sum_{\eta_n<0} w_n\tilde{\eta}_n\psi^l_{n,i,j+1/2}\Bigg]\ ,$$

where

$$c_n = \frac{1}{\frac{|\mu_n|}{\Delta x} + \frac{|\eta_n|}{\Delta y} + \frac{\sigma}{2}}\ ,\qquad \tilde{\mu}_n = c_n\,\frac{|\mu_n|}{\Delta x}\ ,\qquad \tilde{\eta}_n = c_n\,\frac{|\eta_n|}{\Delta y}\ ,\qquad \alpha = \frac{1}{8}\sum_{n=1}^{N(N+2)/2} w_n c_n\ .$$

Likewise, the appropriate diamond difference equations can be combined with the balance equation to obtain expressions for the neutrons emanating from the cell. In each quadrant, with $n = 1,2,\ldots,N(N+2)/8$, we have

$$\psi^{l+1}_{n,i\pm1/2,j} = \left(\tilde{\mu}_n - \tilde{\eta}_n - \frac{1}{2}c_n\sigma_{i,j}\right)\psi^l_{n,i\mp1/2,j} + 2\,\tilde{\eta}_n\,\psi^l_{n,i,j-1/2} + c_n\,Q_{i,j}\ , \qquad \begin{matrix}\eta_n>0\\ \mu_n<0\end{matrix}$$

$$\psi^{l+1}_{n,i\pm1/2,j} = \left(\tilde{\mu}_n - \tilde{\eta}_n - \frac{1}{2}c_n\sigma_{i,j}\right)\psi^l_{n,i\mp1/2,j} + 2\,\tilde{\eta}_n\,\psi^l_{n,i,j+1/2} + c_n\,Q_{i,j}\ , \qquad \begin{matrix}\eta_n<0\\ \mu_n>0\end{matrix}$$

$$\psi^{l+1}_{n,i,j\pm1/2} = \left(\tilde{\eta}_n - \tilde{\mu}_n - \frac{1}{2}c_n\sigma_{i,j}\right)\psi^l_{n,i,j\mp1/2} + 2\,\tilde{\mu}_n\,\psi^l_{n,i-1/2,j} + c_n\,Q_{i,j}\ , \qquad \begin{matrix}\mu_n>0\\ \eta_n>0\end{matrix}$$

$$\psi^{l+1}_{n,i,j\pm1/2} = \left(\tilde{\eta}_n - \tilde{\mu}_n - \frac{1}{2}c_n\sigma_{i,j}\right)\psi^l_{n,i,j\mp1/2} + 2\,\tilde{\mu}_n\,\psi^l_{n,i+1/2,j} + c_n\,Q_{i,j}\ , \qquad \begin{matrix}\mu_n<0\\ \eta_n>0\end{matrix}$$

where the emission density is given by

$$Q_{i,j} = \sigma_{s\,i,j}\ \phi^{l+1}_{i,j} + S_{i,j}\ .$$

Appendix B

The vector equations given in section II may be rewritten in terms of the quantities appearing in foregoing equations. To accomplish this we first order the cell interface angular flux components according which interface (north, south, east, west) they are on and in which of the four quadrants they are trave The outgoing and incoming angular flux vectors may then be written as

$$\Psi^+ = \begin{bmatrix} \mathbf{e}_I \\ \mathbf{n}_I \\ \mathbf{w}_{II} \\ \mathbf{n}_{II} \\ \mathbf{w}_{III} \\ \mathbf{s}_{III} \\ \mathbf{e}_{IV} \\ \mathbf{s}_{IV} \end{bmatrix} \qquad \text{and} \qquad \Psi^- = \begin{bmatrix} \mathbf{w}_I \\ \mathbf{s}_I \\ \mathbf{e}_{II} \\ \mathbf{s}_{II} \\ \mathbf{e}_{III} \\ \mathbf{n}_{III} \\ \mathbf{w}_{IV} \\ \mathbf{n}_{IV} \end{bmatrix}.$$

The elements of Ψ^+ and Ψ^- are themselves each vectors of length $N(N+2)/8$. The ordering of the sub tors is chosen to facilitate data transfer between nearest neighbor nodes. For quadrant $(\mu>0,\eta>0)$ the components are:

$$[\mathbf{n}_I]_n = \psi_{n,i,j+1/2}\ ,$$
$$[\mathbf{s}_I]_n = \psi_{n,i,j-1/2}\ ,$$
$$[\mathbf{e}_I]_n = \psi_{n,i+1/2,j}\ ,$$
$$[\mathbf{w}_I]_n = \psi_{n,i-1/2,j}\ ,$$

and the Roman indices II, III, IV correspond to the quadrants $(\mu<0,\eta>0)$, $(\mu<0,\eta<0)$ a $(\mu>0,\eta<0)$ respectively.

To define v and u, appearing in Eqs.(2) through (6) we group the scalar quantities defined in Appendi into two vectors:

$$\mathbf{v} = \frac{1}{4}\begin{bmatrix} \tilde{\mu} \\ \tilde{\eta} \\ \tilde{\mu} \\ \tilde{\eta} \\ \tilde{\mu} \\ \tilde{\eta} \\ \tilde{\mu} \\ \tilde{\eta} \end{bmatrix} \qquad \text{and} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{c} \\ \mathbf{c} \\ \mathbf{c} \\ \mathbf{c} \\ \mathbf{c} \\ \mathbf{c} \\ \mathbf{c} \\ \mathbf{c} \end{bmatrix},$$

where

$$[\tilde{\mu}]_n = w_n\,\tilde{\mu}_n\ ,$$
$$[\tilde{\eta}]_n = w_n\,\tilde{\eta}_n\ ,$$
$$[\mathbf{c}]_n = c_n\ .$$

Finally, the matrix H, appearing in Eqs.(1) through (6) may be written in the partitioned form

$$\mathbf{H} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{U} & & & & & & \\ \mathbf{L} & \mathbf{D}_2 & & & & & & \\ & & \mathbf{D}_1 & \mathbf{U} & & & & \\ & & \mathbf{L} & \mathbf{D}_2 & & & & \\ & & & & \mathbf{D}_1 & \mathbf{U} & & \\ & & & & \mathbf{L} & \mathbf{D}_2 & & \\ & & & & & & \mathbf{D}_1 & \mathbf{U} \\ & & & & & & \mathbf{L} & \mathbf{D}_2 \end{pmatrix},$$

where the $N(N+2)/8$ nonzero elements of each of the diagonal submatrices are given by:

$$[D_3]_{n\,n'} = \left(\tilde{\mu}_n - \tilde{\eta}_n - \frac{1}{2}c_n\sigma\right)\delta_{n\,n'},$$
$$[D_2]_{n\,n'} = \left(\tilde{\eta}_n - \tilde{\mu}_n - \frac{1}{2}c_n\sigma\right)\delta_{n\,n'},$$
$$[U\,]_{n\,n'} = 2\,\tilde{\eta}_n\,\delta_{n\,n'},$$
$$[L\,]_{n\,n'} = 2\,\tilde{\mu}_n\,\delta_{n\,n'}.$$

DISCLAIMER

DATE
FILMED
01/20/93