NUCLEAR SCIENCE AND ENGINEERING: **115**, 304-319 (1993)

Diffusion-Accelerated Solution of the Two-Dimensional S_n Equations with Bilinear-Discontinuous Differencing

J. E. Morel, J. E. Dendy, Jr., and T. A. Wareing

University of California, Los Alamos National Laboratory, Los Alamos, New Mexico 87545

Received May 24, 1993
Accepted July 15, 1993

Abstract — *A new diffusion synthetic acceleration scheme is developed for solving the two-dimensional S_n equations in x-y geometry with bilinear-discontinuous finite element spatial discretization. This method differs from previous methods in that it is unconditionally efficient for problems with isotropic or weakly anisotropic scattering. Computational results are given that demonstrate this property.*

I. INTRODUCTION

The purpose of this work is to introduce a new diffusion synthetic acceleration (DSA) scheme for solving the two-dimensional S_n equations in x-y geometry with bilinear-discontinuous (BLD) finite element spatial discretization. Our method is the first DSA method for the BLD S_n equations to be unconditionally efficient for problems with isotropic or weakly anisotropic scattering. For instance, Adams and Martin[1] have developed a DSA scheme based on a BLD mixed finite element discretization of the diffusion equation. This diffusion equation has a nonstandard asymmetric form and is not amenable to standard diffusion solution techniques. Adams and Martin were unable to define an unconditionally efficient solution technique for this equation. Although the spectral radius associated with their DSA method is always significantly <1, their method nonetheless becomes inefficient in problems with high scattering ratios because their BLD diffusion equation becomes difficult to solve. Wareing, Larsen, and Adams[2,a] attempted to circumvent this difficulty by developing a DSA method based on a diffusion discretization asymptotically derived from the BLD S_n equations. This asymptotic diffusion equation is much simpler than the BLD diffusion equation of Adams and Martin and has a standard symmetric positive-definite form. Consequently, it can be efficiently solved under all conditions using Dendy's black-box multigrid method.[3] Unfortunately, the DSA method of Wareing, Larsen, and Adams gives a spectral radius that approaches unity in problems with both high scattering ratios and high aspect-ratio spatial zoning.

II. BACKGROUND

Before describing our method in detail, it is appropriate to review the basic concept of synthetic acceleration. Suppose that we wish to solve the following linear system of equations:

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y} \ , \tag{1}$$

where

$\mathbf{A}$ = matrix

$\boldsymbol{x}$ = solution vector

$\boldsymbol{y}$ = source vector.

If the matrix $\mathbf{A}$ is too large to solve directly, a basic iteration scheme is defined by splitting $\mathbf{A}$ into the difference of two matrices:

$$\mathbf{A} = \mathbf{B} - \mathbf{C} \ , \tag{2}$$

[a] The DSA scheme described in this paper is defined for a lumped BLD differencing scheme rather than the standard BLD differencing scheme. However, a standard BLD analogue of this lumped BLD scheme has also been developed by Wareing, Larsen, and Adams, and it exhibits essentially the same behavior as their lumped DSA scheme.

where the matrix $\mathbf{B}$ can be easily inverted. The corresponding iteration scheme can be represented as follows:

$$\boldsymbol{x}^{l+1} = \mathbf{B}^{-1}\mathbf{C}\boldsymbol{x}^{l} + \mathbf{B}^{-1}\boldsymbol{y} \ , \tag{3}$$

where l is the iteration index. Let us define the error associated with the iterate at step $l+1$ as follows:

$$\boldsymbol{\epsilon}^{l+1} = \boldsymbol{x} - \boldsymbol{x}^{l+1} \ , \tag{4}$$

where $\boldsymbol{x}$ denotes the solution to Eq. (1). It is not difficult to demonstrate that this error satisfies the following equation:

$$\mathbf{A}\boldsymbol{\epsilon}^{l+1} = \boldsymbol{R}^{l+1} \ , \tag{5}$$

where the residual is given by

$$\boldsymbol{R}^{l+1} = \boldsymbol{y} - \mathbf{A}\boldsymbol{x}^{l+1} \ . \tag{6}$$

In principle, one can obtain the error by solving Eq. (5), but this equation is just as difficult to solve as Eq. (1). Consequently, an exact solution of this error equation is not practical. However, it is conceivable that obtaining an approximate solution to the error equation could yield an error estimate that would significantly improve the accuracy of $\boldsymbol{x}^{l}$ and thereby accelerate the convergence of the iterative process. This is the basic theme of synthetic acceleration methods. For example, in the simplest synthetic acceleration scheme, one would approximate $\mathbf{A}$ in Eq. (5) with a low-rank operator:

$$\langle \mathbf{A} \rangle \langle \boldsymbol{\epsilon} \rangle^{l+1} = \langle \boldsymbol{R} \rangle^{l+1} \ , \tag{7}$$

where the angle brackets denote a low-rank approximation. The low-rank residual appearing in Eq. (7) would be obtained from the full-rank residual by means of a projection:

$$\langle \boldsymbol{R} \rangle^{l+1} = \mathbf{P}\boldsymbol{R}^{l+1} \ , \tag{8}$$

where $\mathbf{P}$ is an operator that projects from the full-rank space to the low-rank space. The low-rank error equation would be solved to obtain the low-rank error estimate:

$$\langle \boldsymbol{\epsilon} \rangle^{l+1} = \langle \mathbf{A} \rangle^{-1} \langle \boldsymbol{R} \rangle^{l+1} \ . \tag{9}$$

A full-rank error estimate would be obtained from the low-rank error estimate by interpolation:

$$\boldsymbol{\epsilon}^{l+1} = \mathbf{T}\langle \boldsymbol{\epsilon} \rangle^{l+1} \ , \tag{10}$$

where $\mathbf{T}$ is an interpolation operator. The error estimate would then be added to the iterate at step $l+1$ to obtain an improved or accelerated iterate.

To summarize, an accelerated iteration is carried out as follows:

1. A basic iteration is performed:

$$\boldsymbol{x}^{l+1/2} = \mathbf{B}^{-1}\mathbf{C}\boldsymbol{x}^{l} + \mathbf{B}^{-1}\boldsymbol{y} \ , \tag{11}$$

where the index $l + \frac{1}{2}$ is used in anticipation of the acceleration step.

2. The residual associated with the basic iterate is calculated:

$$\boldsymbol{R}^{l+1/2} = \boldsymbol{y} - \mathbf{A}\boldsymbol{x}^{l+1/2} \ . \tag{12}$$

3. The low-rank residual is calculated from the high-rank residual via projection:

$$\langle \boldsymbol{R} \rangle^{l+1/2} = \mathbf{P}\boldsymbol{R}^{l+1/2} \ . \tag{13}$$

4. The low-rank error equation is solved:

$$\langle \boldsymbol{\epsilon} \rangle^{l+1/2} = \langle \mathbf{A} \rangle^{-1} \langle \boldsymbol{R} \rangle^{l+1/2} \ . \tag{14}$$

5. The high-rank error estimate is calculated from the low-rank estimate via interpolation:

$$\boldsymbol{\epsilon}^{l+1/2} = \mathbf{T}\langle \boldsymbol{\epsilon} \rangle^{l+1/2} \ . \tag{15}$$

6. The high-rank error estimate is added to the unaccelerated iterate to obtain the accelerated iterate:

$$\boldsymbol{x}^{l+1} = \boldsymbol{x}^{l+1/2} + \boldsymbol{\epsilon}^{l+1/2} \ . \tag{16}$$

A synthetic acceleration scheme will be effective if two conditions are met. First, the low-rank equation must accurately estimate the errors that are poorly attenuated by the basic iteration scheme. Second, the low-rank operator must not significantly overestimate the errors that are strongly attenuated by the basic iteration scheme.

For the specific case of DSA, $\mathbf{A}$ is the transport operator, $\mathbf{B}$ is the sum of the streaming and removal operators, $\mathbf{C}$ is the scattering operator, and $\langle \mathbf{A} \rangle$ is the diffusion operator. The scattering may be anisotropic, but only the isotropic component of the angular flux is accelerated. The projection operator $\mathbf{P}$ maps an angular flux function to its isotropic or P_0 moment, and the interpolation operator $\mathbf{T}$ maps a P_0 moment to the P_0 angular flux expansion. Diffusion synthetic acceleration is effective in terms of the error reduction per iteration with isotropic or weakly anisotropic scattering for two reasons. First, the errors that are poorly attenuated by the transport sweep are diffusive and thus accurately calculated with the diffusion operator. Second, the nondiffusive errors that are strongly attenuated by the transport sweep are underestimated by the diffusion equation.

More complicated synthetic acceleration methods have multiple levels of acceleration. For instance, the low-rank operator used in the Eq. (14) may still be too large or complicated to be solved efficiently using a direct or unaccelerated iterative method. Attempts to further simplify the low-rank operator may be fruitless because an operator that is simple enough to solve easily may be too simple to provide an effective approximation to the high-rank operator. One possible solution is to develop an accelerated iteration scheme for inverting the low-rank operator. This leads to a scheme with two acceleration levels. Any number of levels is possible. The low-rank operator on each level approximates the higher rank operator on the level above it. Such schemes are also referred to as multigrid schemes. In multigrid terminology, the high-rank and low-rank operators are called the "fine-grid" and the "coarse-grid" operators. In general, optimum efficiency in a multilevel scheme is obtained by fully solving the error equation only on the lowest level. The error equations on the intermediate levels are solved approximately.

III. THE NEW DIFFUSION SYNTHETIC ACCELERATION METHOD

The DSA method that we have developed can be thought of as a multilevel synthesis of the Adams-Martin and Wareing-Larsen-Adams methods. In particular, at the first level, the BLD S_n iterations are accelerated with a slightly modified version of the Adams-Martin BLD diffusion equation. At the second level, the BLD diffusion iterations are accelerated with a bilinear-continuous (BLC) diffusion equation that is equivalent to the Wareing-Larsen-Adams asymptotic diffusion equation. Finally, at the third level, the BLC diffusion iterations are accelerated with Dendy's black-box multigrid algorithm. The overall DSA algorithm resulting from this multilevel approach is unconditionally efficient. In particular, a homogeneous infinite-medium Fourier analysis performed by Adams and Martin for the first iteration level gives a worst-case spectral radius of ~0.5. We have performed a similar Fourier analysis for the second iteration level that also gives a worst-case spectral radius of ~0.5. Dendy's algorithm, which is used on the third level of acceleration, has previously been shown to have a spectral radius of ~0.1 for most realistic problems. Unconditional efficiency for the overall DSA scheme follows directly from the unconditional efficiency achieved on each level.

To be fully consistent with the Fourier analyses that we have performed, the BLD and BLC diffusion solutions used in our acceleration scheme should be iterated to convergence. However, it is much more efficient to accept the solutions to these low-rank equations after a fixed number of iterations. In particular, the BLD solution is accepted after three iterations, and the BLC solution is accepted after one iteration. This strategy gives our DSA scheme a true multigrid character. If the BLD diffusion equation were fully solved, our new DSA scheme would be more properly characterized as the DSA scheme of Adams and Martin[1] with a new technique for solving the associated BLD diffusion equation.

To facilitate a more detailed description of our acceleration technique, we now describe certain basic aspects of the spatial differencing schemes used on each level. Both the BLD S_n differencing scheme and the BLD diffusion differencing scheme have unknowns at the corners of each spatial cell. These locations are shown in Fig. 1. The BLD S_n scheme has an angular flux unknown at each location while the BLD diffusion scheme has a scalar flux unknown at each location. The BLC diffusion differencing scheme has a scalar flux unknown at each vertex of the mesh. This is illustrated in Fig. 2.

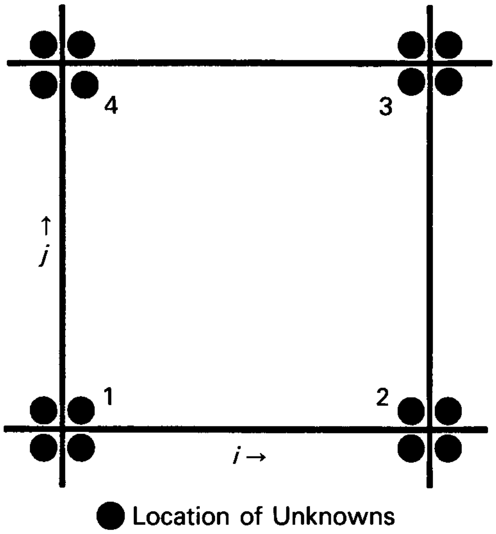

Fig. 1. Spatial mesh and corner indexing for BLD S_n and diffusion.

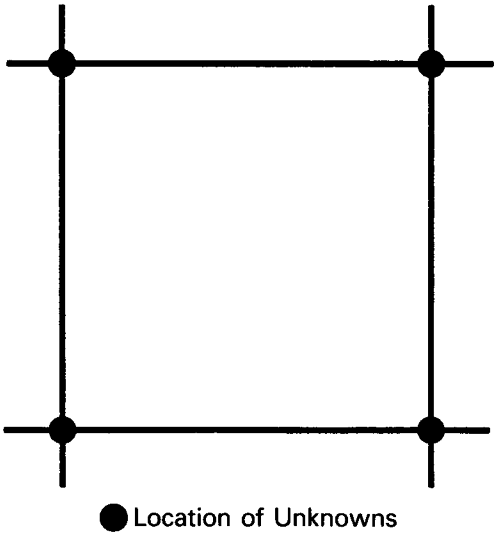

Fig. 2. Spatial mesh for BLC diffusion.

The BLD S_n differencing scheme that we use is identical (after a similarity transformation) to that used by Adams and Martin,[1] and the BLC diffusion differencing scheme that we use is identical to that used by Wareing, Larsen, and Adams.[2] However, the BLD diffusion differencing scheme that we use is equivalent only on the interior of the mesh to the Adams-Martin

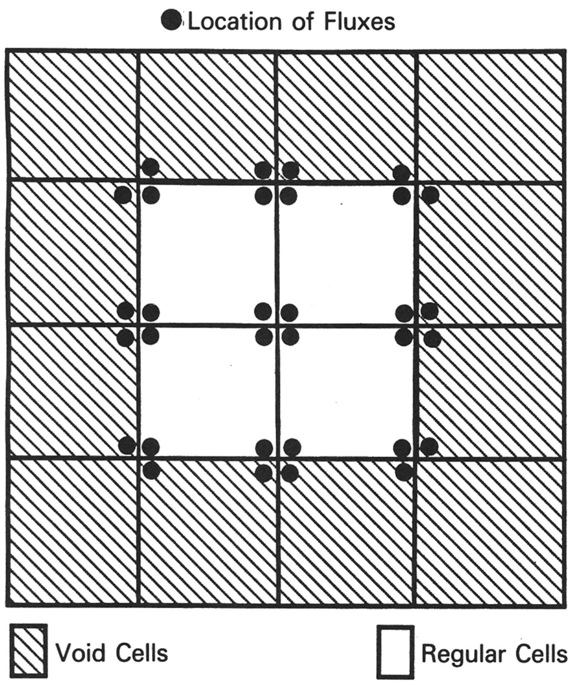

Fig. 3. BLD diffusion void fluxes.

scheme. Our scheme has additional scalar flux unknowns along the outer boundaries of the mesh, as illustrated in Fig. 3. These fluxes are referred to as "void" fluxes because they are associated with cells that are outside of the mesh. Although the void fluxes couple to the regular fluxes on the outer boundaries of the mesh, the solutions obtained for all of the regular fluxes are identical to those obtained with the Adams-Martin scheme. Since only the regular fluxes are explicitly used to estimate the BLD S_n iterate errors, our BLD diffusion scheme is completely equivalent (for acceleration purposes) to that of Adams and Martin. However, for reasons explained below, we can solve our BLD diffusion equations more efficiently than we can solve those of Adams and Martin.

Wareing, Larsen, and Adams asymptotically derived their BLC diffusion equations from the BLD S_n equations. However, they substituted standard Marshak boundary conditions for the asymptotic conditions because the latter are not suitable for acceleration purposes. We found that the BLC diffusion equations can also be derived asymptotically from the Adams-Martin BLD diffusion equations. It was this result that first suggested to us that the BLC diffusion equations might be an effective low-rank approximation to the Adams-Martin BLD diffusion equations. As expected, we found that the asymptotic boundary conditions for the BLC equations are not suitable for acceleration purposes, so we substituted standard Marshak conditions. Unexpected difficulties arose on the outer boundaries of the mesh when we attempted to accelerate the iterative convergence of the Adams-Martin BLD diffusion scheme with the BLC diffusion scheme. Specifically, we could not define a projection operator from the BLD mesh to the BLC mesh that results in an exact calculation of BLD errors that are continuous at each cell vertex. One would expect an exact calculation because such errors can be exactly represented on the BLC mesh.

We were able to eliminate these difficulties after recognizing that a simple relationship exists between the Adams-Martin BLD equations and the BLC equations on the interior of the mesh. Specifically, the interior-mesh BLC diffusion equations can be derived from the interior-mesh Adams-Martin BLD diffusion equations in the following manner:

1. Assume that four BLD fluxes associated with each vertex of the mesh are identical. This is equivalent to requiring continuity of the scalar flux solution and leaves a single scalar flux unknown at each vertex.

2. Obtain an equation for each "continuous" vertex flux by summing the BLD equations for the four scalar fluxes associated with that vertex.

Having recognized the existence of this relationship on the interior of the mesh, we postulated that it should also apply on the mesh boundaries. Our BLD diffusion scheme was obtained by modifying the Adams-Martin scheme to achieve this property. Specifically, if one follows the derivation procedure defined earlier using our BLD diffusion equations and if one carries out the procedure at all cell vertices (including those on the outer mesh boundaries), one obtains the BLC diffusion equations with Marshak boundary conditions. Note from Fig. 3 that only three BLC fluxes exist at each of the four corner vertices on the outer boundaries of the mesh. Hence, only three fluxes are summed at these points. This summing process defines the projection from the BLD diffusion mesh to the BLC diffusion mesh. Specifically, a residual vector on the BLD diffusion mesh is projected onto the BLC diffusion mesh by summing the residuals about each cell vertex to obtain a single vertex residual. The difficulties that we encountered with residual projection on the outer boundaries when trying to accelerate the Adams-Martin BLD diffusion scheme with the BLC diffusion scheme do not arise when our scheme is substituted for the Adams-Martin scheme.

The basic iteration scheme used for our BLD diffusion equations is a line-Jacobi scheme.[4] One complete line-Jacobi iteration consists of an x-line-Jacobi iteration followed by a y-line-Jacobi iteration. An x line consists of all of the fluxes having the same y coordinate, and a y line consists of all of the fluxes having the same x coordinate. An x line is illustrated in Fig. 4. The iteration equations for each x-line iteration are constructed from the full equations by lagging the coupling to all other fluxes not in the x line. The lagged fluxes

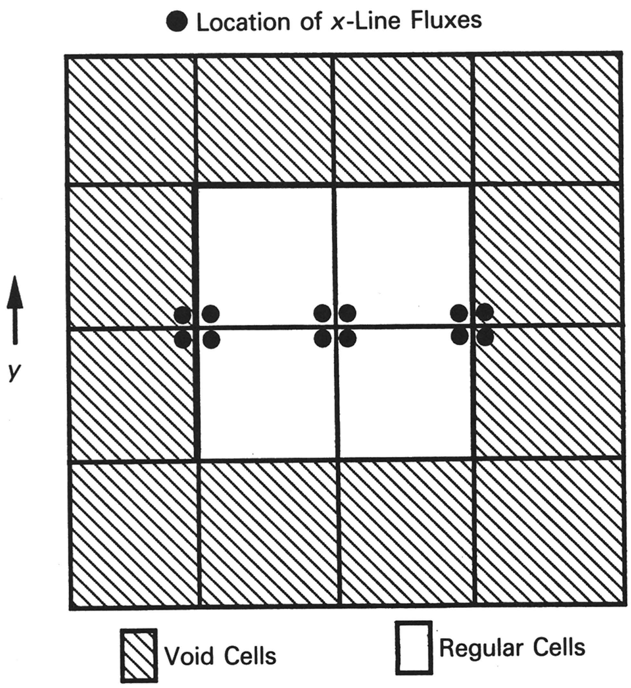

Fig. 4. Fluxes in one x line.

are held at iterate values associated with the beginning of the x-line-Jacobi iteration, and the fluxes in all x lines are calculated with the same set of lagged values. A y-line-Jacobi iteration is analogously defined. The lagged fluxes are updated between the x-line-Jacobi and the y-line-Jacobi iterations. We tried other line relaxation schemes including the Gauss-Seidel and red-black schemes. However, the line-Jacobi scheme performs best.

The accelerated iteration scheme used for the BLC equations is Dendy's black-box multigrid scheme.[3] This is a true multigrid scheme that uses diffusion operators defined on coarse spatial grids as low-rank operators. Dendy has implemented his scheme in a FORTRAN package that is available from him. We employ this package without modification. The package solves any nine-point diffusion operator on a logically rectangular mesh.

We now give a final and more detailed description of our DSA method. One accelerated S_n iteration proceeds in the following manner:

1. The BLD S_n scattering sources are calculated, and the sweep equations are solved.

2. The isotropic moments of the S_n residuals are calculated for use in the BLD diffusion equation. Note that a spatial projection from the S_n to BLD diffusion mesh is not necessary since these schemes use the same spatial mesh.

3. A line-Jacobi iteration is performed on the BLD diffusion equations.

4. The BLD diffusion residuals are calculated and projected onto the BLC diffusion mesh by summing the four BLD residuals associated with each vertex to obtain a single vertex residual.

5. One V-cycle of Dendy's black-box multigrid algorithm is performed on the BLC diffusion equations.

6. The BLC diffusion iterate is interpolated onto the BLD diffusion mesh by assuming continuity; i.e., the four BLD mesh values needed for each corner are set equal to the single BLC vertex value.

7. The interpolated BLC iterate is then added to the BLD iterate to obtain the "accelerated" BLD diffusion iterate.

8. Steps 3 through 7 are repeated twice, resulting in three accelerated BLD diffusion iterations.

9. The BLD diffusion iterate is added to the BLD S_n scalar flux iterate to obtain the accelerated S_n scalar fluxes. This completes one accelerated S_n iteration.

All of the discretized equations used in our method are derived in the appendixes. The BLD S_n equations are derived in Appendix A, the BLD diffusion equations are derived in Appendix B, and the BLC diffusion equations are derived in Appendix C.

IV. COMPUTATIONAL RESULTS

In this section, we give computational results that demonstrate that our DSA method remains efficient for problems that cause the Adams-Martin[1] and Wareing-Larsen-Adams[2] methods to become inefficient. We have performed three sets of calculations.

The first set of calculations demonstrates the effectiveness of our DSA scheme in terms of error reduction per iteration. The geometry for the first set corresponds to a homogeneous rectangular region illustrated in Fig. 5. This region has isotropic scattering, a scattering ratio of unity, and a constant isotropic distributed source. The rectangle has reflective boundaries on the bottom and left sides and vacuum boundaries on the top and right sides. There are 25 cells along the x axis and 25 cells along the y axis. In each calculation, there is a single x-axis cell width and a single y-axis cell width, but these two widths are not necessarily the same. These widths vary between calculations. All of the calculations in this set were performed with an S_4 quadrature set.

The second set of calculations characterizes the overall efficiency of our scheme as a function of scattering ratio. The geometry for this set is identical to that of the first set. Both the x-axis and y-axis cell widths are fixed at 1.0 mean free path (mfp), and an S_4 quadrature set is used in all of the calculations in this set. The scattering ratio is varied from 1.0 to 0.1, and each calculation is performed once without acceleration and once with acceleration.

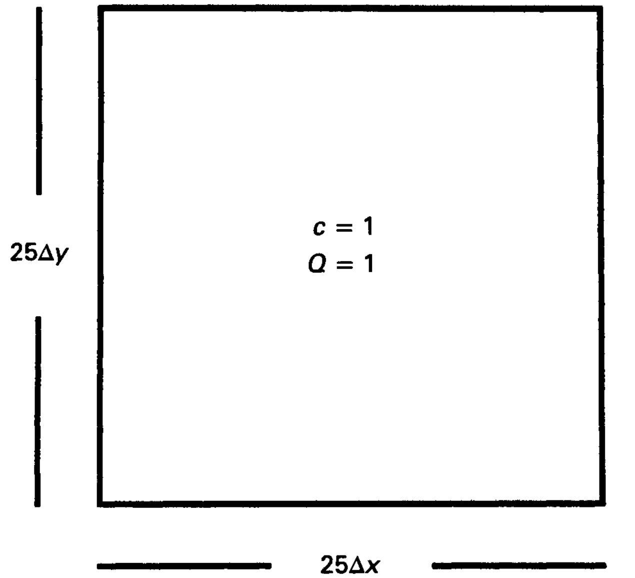

Fig. 5. Geometry for problem set 1.

The third set of calculations demonstrates the effectiveness of our scheme for inhomogeneous problems. The geometry for this set is illustrated in Fig. 6. It consists of a rectangular region that is 50 cm in length and width with an inner region 10 cm in length and width. The rectangle has reflective boundaries on the bottom and left sides and vacuum boundaries on the top and right sides. Both the inner and outer regions have a total cross section of 1.0 cm^{-1} and a uniform isotropic distributed source. The inner region has a scattering ratio of 1.0 while the outer region has a scattering ratio of 0.95. The number of spatial cells varies between calculations, but the x-axis and y-axis cell widths are always equal. All of the calculations in this set were performed with an S_8 quadrature set.

All calculations were performed on a Cray-YMP computer single processor. The scalar flux in every calculation was subject to a pointwise relative convergence criterion of 10^{-4}. The spectral radius for each calculation was estimated according to the following expression:

$$\rho = \frac{\|\phi_0^L - \phi_0^{L-1}\|}{\|\phi_0^{L-1} - \phi_0^{L-2}\|} , \qquad (17)$$

where

ϕ_0^l = vector of scalar fluxes obtained from the l'th iteration

L = index of the last iteration performed in the calculation

$\|\boldsymbol{x}\|$ = standard Euclidian norm of $\boldsymbol{x}$.

The results for the first set of calculations are given in Table I. The information appearing in this table consists of the x-axis cell width, the y-axis cell width, the total CPU time, the CPU time spent solving the DSA equations, the number of iterations required to converge the S_n solution, and the estimated spectral radius. It can be seen from Table I that our method requires no more than eight iterations for optically thick problems with a scattering ratio of 1, including problems with high aspect-ratio spatial zoning. Optically thick problems with a scattering ratio of 1 cause difficulty

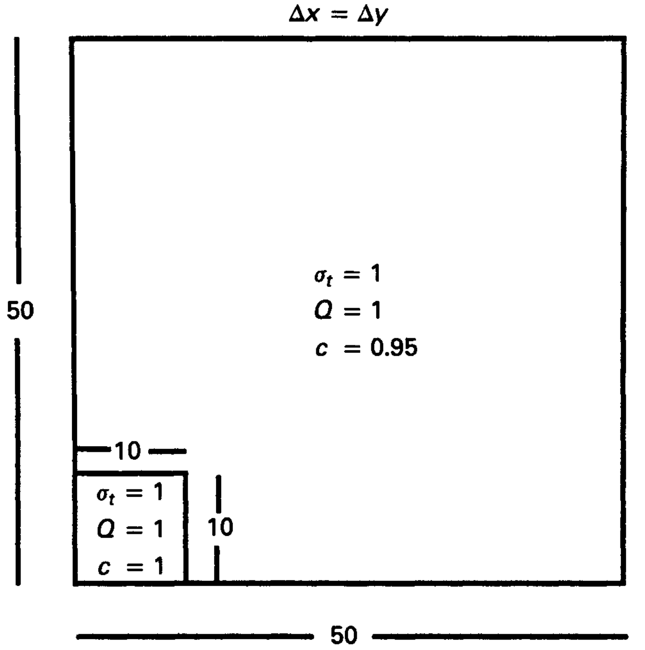

Fig. 6. Geometry for problem set 3. All quantities are in cgs units.

TABLE I

Results for Problem Set 1

Δx (mfp)	Δy (mfp)	Total CPU (s)	DSA CPU (s)	Iterations	Spectral Radius
0.1	0.1	0.60	0.31	6	0.24
0.1	1.0	0.79	0.39	8	0.38
0.1	5.0	0.60	0.31	6	0.26
0.1	10.0	0.60	0.31	6	0.21
0.1	100.0	0.60	0.31	6	0.21
1.0	1.0	0.79	0.39	8	0.45
1.0	5.0	0.79	0.39	8	0.45
1.0	10.0	0.79	0.39	8	0.42
1.0	100.0	0.79	0.39	8	0.42
5.0	5.0	0.60	0.31	6	0.42
5.0	10.0	0.60	0.31	6	0.42
5.0	100.0	0.60	0.31	6	0.42
10.0	10.0	0.51	0.26	5	0.29
10.0	100.0	0.51	0.26	5	0.29
100.0	100.0	0.51	0.26	5	0.29

for the Adams-Martin method, and problems with high aspect-ratio spatial zoning cause difficulty for the Wareing-Larsen-Adams method. About one-half of the total CPU time is spent solving the DSA equations. This fraction will rapidly approach zero as the quadrature order is increased.

The results for the second set of calculations are given in Table II. The information appearing in the table consists of the scattering ratio, the number of iterations required to converge the S_n solution, and the total CPU time required. Each problem is performed once with acceleration and once without acceleration. It can be seen from Table II that our DSA scheme is extremely efficient for problems with scattering ratios near unity. For instance, when the scattering ratio is unity, DSA reduces the total CPU time by a factor of ~235. The scheme remains efficient for all scattering ratios greater than ~0.3. The scattering ratio at which DSA becomes inefficient will rapidly approach zero as the quadrature order is increased.

The results for the third set of calculations are given in Table III. The information appearing in this table consists of the mesh size, the cell width, the total CPU time, the CPU time spent solving the DSA equations, the number of iterations required to converge the S_n solution, and the estimated spectral radius. With the exception of the CPU times, the results obtained for the third set of calculations are very similar to those obtained for the first set of calculations. For all but the meshes with the fewest cells, the CPU time spent solving the DSA equations is a much smaller fraction of the total CPU time. This occurs simply because an S_8 quadrature set was used in the third set of calculations while an S_4 quadrature set was used in the first set of calculations. The CPU time spent solving the DSA equations relative to the total CPU time dramatically increases as the number of cells decreases. This is a result of the overhead associated with solving the acceleration equations on several levels. When the work done on each level becomes very small, the overhead becomes dominant. Perhaps the most important result from this set of calculations is that there appears to be no significant degradation of the method due to material inhomogeneities.

V. CONCLUSIONS AND FUTURE WORK

We have tested our method on a wide variety of problems in addition to those presented here. We have found our method to be highly efficient for all problems with high scattering ratios and moderately anisotropic scattering. To our knowledge, no previous DSA scheme for the two-dimensional *x-y* S_n equations with BLD spatial differencing (or any other type of advanced differencing) has been unconditionally efficient. In the near future, we will show that our BLD diffusion differencing scheme can also be used to accelerate the two-dimensional *x-y* S_n equations with linear-bilinear nodal differencing.[5] It appears that our method can easily be extended to *r-z* geometry with rectangular meshes, but this conjecture remains to be proven. We hope to eventually investigate both nonorthogonal-mesh generalizations and three-dimensional generalizations of our method. The multilevel approach appears to be both powerful and widely applicable. For instance, it has also been used to efficiently solve nonstandard diffusion equations on nonorthogonal two-dimensional *r-z* quadrilateral meshes.[6]

APPENDIX A

THE BILINEAR-DISCONTINUOUS S_n EQUATIONS

In this appendix, we derive the BLD differencing scheme for the *x-y* geometry S_n equations given by

TABLE II

Results for Problem Set 2

Scattering Ratio	Number of Iterations		Total CPU (s)	
	Unaccelerated	Accelerated	Unaccelerated	Accelerated
1.0	3764	8	187.0	0.79
0.9	89	7	4.42	0.70
0.8	43	7	2.14	0.70
0.7	27	6	1.35	0.60
0.6	20	5	1.00	0.51
0.5	15	5	0.75	0.51
0.4	12	5	0.60	0.51
0.3	9	4	0.45	0.41
0.2	7	4	0.35	0.41
0.1	5	3	0.24	0.32

TABLE III

Results for Problem Set 3

Mesh Size	Δx and Δy (mfp)	Total CPU (s)	DSA CPU (s)	Iterations	Spectral Radius
5 × 5	10.0	0.11	0.07	6	0.19
10 × 10	5.00	0.39	0.18	8	0.34
15 × 15	3.33	0.86	0.27	10	0.45
20 × 20	2.50	1.40	0.36	10	0.45
25 × 25	2.00	1.90	0.43	9	0.45
30 × 30	1.67	2.66	0.55	9	0.45
35 × 35	1.43	3.60	0.73	9	0.43
40 × 40	1.25	4.11	0.77	8	0.41
45 × 45	1.11	5.24	1.03	8	0.38
50 × 50	1.00	6.28	1.08	8	0.38
60 × 60	0.83	7.43	1.28	7	0.36
70 × 70	0.71	10.8	1.89	7	0.33
80 × 80	0.63	13.9	2.30	7	0.31
90 × 90	0.56	17.5	2.76	7	0.29
100 × 100	0.50	21.5	3.31	7	0.28
120 × 120	0.42	30.8	4.61	7	0.25
140 × 140	0.36	36.1	5.55	6	0.23
160 × 160	0.31	40.9	7.04	6	0.22
180 × 180	0.28	59.8	8.61	6	0.21
200 × 200	0.25	73.3	11.0	6	0.21
300 × 300	0.17	163.5	23.2	6	0.21

$$\mu_m \frac{\partial \psi_m(x,y)}{\partial x} + \eta_m \frac{\partial \psi_m(x,y)}{\partial y} + \sigma_t(x)\psi_m(x,y) = \sigma_{s0}(x,y)\phi(x,y) + Q(x,y) \ ,$$
$$1 \le m \le M,\ x_{1/2} < x < x_{I+1/2},\ y_{1/2} < y < y_{J+1/2} \ , \tag{A.1}$$

where

$$\phi(x,y) = \sum_{m=1}^{M} \psi_m(x,y)w_m \ , \quad M = \frac{n(n+2)}{2} \ . \tag{A.2}$$

The boundary conditions to Eq. (A.1) are given by

$$\psi_m(x_{1/2},y) = f_m(x_{1/2},y) \ , \quad \mu_m > 0,\ y_{1/2} < y < y_{J+1/2} \ , \tag{A.3}$$

$$\psi_m(x_{I+1/2},y) = f_m(x_{I+1/2},y) \ , \quad \mu_m < 0,\ y_{1/2} < y < y_{J+1/2} \ , \tag{A.4}$$

$$\psi_m(x,y_{1/2}) = f_m(x,y_{1/2}) \ , \quad \eta_m > 0,\ x_{1/2} < x < x_{I+1/2} \ , \tag{A.5}$$

and

$$\psi_m(x,y_{J+1/2}) = f_m(x,y_{J+1/2}) \ , \quad \eta_m < 0,\ x_{1/2} < x < x_{I+1/2} \ . \tag{A.6}$$

This problem is defined for the spatial mesh given in Fig. 1, where the material properties are assumed to be constant within each spatial cell.

First, we define a set of basis functions $B_{k,i,j}(x)$ (k = 1,4), which describe the behavior of the flux within the i,j'th cell. Here, k represents the corners of the i,j'th cell as shown in Fig. 1. We use piecewise-linear basis functions given by

$$B_{1,i,j}(x,y) = \left(1 - \frac{x - x_{i-1/2}}{\Delta x_i}\right)\left(1 - \frac{y - y_{j-1/2}}{\Delta y_j}\right) \ , \tag{A.7}$$

$$B_{2,i,j}(x,y) = \left(\frac{x - x_{i-1/2}}{\Delta x_i}\right)\left(1 - \frac{y - y_{j-1/2}}{\Delta y_j}\right) \ , \tag{A.8}$$

$$B_{3,i,j}(x,y) = \left(\frac{x - x_{i-1/2}}{\Delta x_i}\right)\left(\frac{y - y_{j-1/2}}{\Delta y_j}\right) \ , \tag{A.9}$$

and

$$B_{4,i,j}(x,y) = \left(1 - \frac{x - x_{i-1/2}}{\Delta x_i}\right)\left(\frac{y - y_{j-1/2}}{\Delta y_j}\right) \ , \tag{A.10}$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and $\Delta y_j = y_{j+1/2} - y_{j-1/2}$.

We begin by operating on Eq. (18) by

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} B_{k,i,j}(x,y)(\cdot)\,dx\,dy \ ,$$

where $1 \le i \le I$ and $1 \le j \le J$. Using Green's theorem, we obtain

$$\begin{aligned}
&\int_{x_{i-1/2}}^{x_{i+1/2}} \eta_m [B_{k,i,j}(x,y_{j+1/2})\psi_m(x,y_{j+1/2}) \\
&\qquad - B_{k,i,j}(x,y_{j-1/2})\psi_m(x,y_{j-1/2})]\,dx \\
&\quad + \int_{y_{j-1/2}}^{y_{j+1/2}} \mu_m [B_{k,i,j}(x_{i+1/2},y)\psi_m(x_{i+1/2},y) \\
&\qquad - B_{k,i,j}(x_{i-1/2},y)\psi_m(x_{i-1/2},y)]\,dy \\
&\quad + \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \psi_m(x,y) \\
&\quad \times \left[\mu_m \frac{\partial B_{k,i,j}(x,y)}{\partial x} + \eta_m \frac{\partial B_{k,i,j}(x,y)}{\partial y}\right] dx\,dy \\
&\quad + \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \sigma_{t,i,j} B_{k,i,j}(x,y)\psi_m(x,y)\,dx\,dy \\
&= \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} B_{k,i,j}(x,y) \\
&\quad \times [\sigma_{s0,i,j}\phi(x,y) + Q(x,y)]\,dx\,dy \ . \qquad \text{(A.11)}
\end{aligned}$$

Next, we introduce the following approximations:

$$\psi_m(x,y) = \sum_{k=1}^{4} B_{k,i,j}(x,y)\Psi_{m,k,i,j} \ , \qquad \text{(A.12)}$$

$$\psi_m(x_{i-1/2},y) = \sum_{k=1}^{4} B_{k,i,j}(x_{i-1/2},y)\Psi^L_{m,k,i,j} \ , \qquad \text{(A.13)}$$

$$\psi_m(x_{i+1/2},y) = \sum_{k=1}^{4} B_{k,i,j}(x_{i+1/2},y)\Psi^R_{m,k,i,j} \ , \qquad \text{(A.14)}$$

$$\psi_m(x,y_{j-1/2}) = \sum_{k=1}^{4} B_{k,i,j}(x,y_{j-1/2})\Psi^B_{m,k,i,j} \ , \qquad \text{(A.15)}$$

$$\psi_m(x,y_{j+1/2}) = \sum_{k=1}^{4} B_{k,i,j}(x,y_{j+1/2})\Psi^T_{m,k,i,j} \ , \qquad \text{(A.16)}$$

$$\phi(x,y) = \sum_{k=1}^{4} B_{k,i,j}(x,y)\Phi_{k,i,j} \ , \qquad \text{(A.17)}$$

and

$$Q(x,y) = \sum_{k=1}^{4} B_{k,i,j}(x,y)Q_{k,i,j} \ , \qquad \text{(A.18)}$$

where

$$\Psi^L_{m,1,i,j} = \begin{cases} f^L_{m,1,j} \ , & \mu_m > 0,\ i = 1, \\ \psi_{m,2,i-1,j} \ , & \mu_m > 0,\ 2 \le i \le I, \\ & \qquad 1 \le j \le J \ , \\ \psi_{m,1,i,j} \ , & \mu_m < 0,\ 1 \le i \le I \ , \end{cases} \qquad \text{(A.19)}$$

$$\Psi^L_{m,4,i,j} = \begin{cases} f^L_{m,4,j} \ , & \mu_m > 0,\ i = 1 \ , \\ \psi_{m,3,i-1,j} \ , & \mu_m > 0,\ 2 \le i \le I, \\ & \qquad 1 \le j \le J \ , \\ \psi_{m,4,i,j} \ , & \mu_m < 0,\ 1 \le i \le I \ , \end{cases} \qquad \text{(A.20)}$$

$$\Psi^R_{m,2,i,j} = \begin{cases} \psi_{m,2,i,j} \ , & \mu_m > 0,\ 1 \le i \le I \ , \\ \psi_{m,1,i+1,j} \ , & \mu_m < 0,\ 1 \le i \le I-1, \\ & \qquad 1 \le j \le J \ , \\ f^R_{m,2,j} \ , & \mu_m < 0,\ i = I \ , \end{cases} \qquad \text{(A.21)}$$

$$\Psi^R_{m,3,i,j} = \begin{cases} \psi_{m,3,i,j} \ , & \mu_m > 0,\ 1 \le i \le I \ , \\ \psi_{m,4,i+1,j} \ , & \mu_m < 0,\ 1 \le i \le I-1, \\ & \qquad 1 \le j \le J \ , \\ f^R_{m,3,j} \ , & \mu_m < 0,\ i = I \ , \end{cases} \qquad \text{(A.22)}$$

$$\Psi^B_{m,1,i,j} = \begin{cases} f^B_{m,1,i} \ , & \eta_m > 0,\ j = 1 \ , \\ \psi_{m,4,i,j-1} \ , & \eta_m > 0,\ 2 \le j \le J, \\ & \qquad 1 \le i \le I \ , \\ \psi_{m,1,i,j} \ , & \eta_m < 0,\ 1 \le j \le J \ , \end{cases} \qquad \text{(A.23)}$$

$$\Psi^B_{m,2,i,j} = \begin{cases} f^B_{m,2,i} \ , & \eta_m > 0,\ j = 1 \ , \\ \psi_{m,3,i,j-1} \ , & \eta_m > 0,\ 2 \le j \le J, \\ & \qquad 1 \le i \le I \ , \\ \psi_{m,2,i,j} \ , & \eta_m < 0,\ 1 \le j \le J \ , \end{cases} \qquad \text{(A.24)}$$

$$\Psi^T_{m,3,i,j} = \begin{cases} \psi_{m,3,i,j} \ , & \eta_m > 0,\ 1 \le j \le J \ , \\ \psi_{m,2,i,j+1} \ , & \eta_m < 0,\ 1 \le j \le J-1, \\ & \qquad 1 \le i \le I \ , \\ f^T_{m,3,i} \ , & \eta_m < 0,\ j = J \ , \end{cases} \qquad \text{(A.25)}$$

and

$$\Psi^T_{m,4,i,j} = \begin{cases} \psi_{m,4,i,j} \ , & \eta_m > 0,\ 1 \le j \le J \ , \\ \psi_{m,1,i,j+1} \ , & \eta_m < 0,\ 1 \le j \le J-1, \\ & \qquad 1 \le i \le I \ , \\ f^T_{m,4,i} \ , & \eta_m < 0,\ j = J \ . \end{cases} \qquad \text{(A.26)}$$

Here, $f^L_{m,k,j}$, $f^R_{m,k,j}$, $f^B_{m,k,j}$, and $f^T_{m,k,j}$ are the discrete incident angular fluxes on the left, right, bottom, and top boundaries, respectively.

Performing the integrals, we obtain the following equations, which are given in matrix form:

$$\frac{\mu_m}{\Delta x_i}\mathbf{U}\begin{bmatrix}\Psi^L_{m,1,i,j}\\ \Psi^R_{m,2,i,j}\\ \Psi^R_{m,3,i,j}\\ \Psi^L_{m,4,i,j}\end{bmatrix} + \frac{\eta_m}{\Delta y_j}\mathbf{N}\begin{bmatrix}\Psi^B_{m,1,i,j}\\ \Psi^B_{m,2,i,j}\\ \Psi^T_{m,3,i,j}\\ \Psi^T_{m,4,i,j}\end{bmatrix}$$

$$+\left(\frac{\mu_m}{\Delta x_i}\mathbf{L} + \frac{\eta_m}{\Delta y_j}\mathbf{K} + \sigma_{t,i,j}\mathbf{M}\right)\begin{bmatrix}\Psi_{m,1,i,j}\\ \Psi_{m,2,i,j}\\ \Psi_{m,3,i,j}\\ \Psi_{m,3,i,j}\end{bmatrix}$$

$$= \sigma_{s0,i,j}\mathbf{M}\begin{bmatrix}\Phi_{m,1,i,j}\\ \Phi_{m,2,i,j}\\ \Phi_{m,3,i,j}\\ \Phi_{m,3,i,j}\end{bmatrix} + \mathbf{M}\begin{bmatrix}q_{1,i,j}\\ q_{2,i,j}\\ q_{3,i,j}\\ q_{4,i,j}\end{bmatrix} \quad . \qquad \text{(A.27)}$$

Here, we have defined

$$\mathbf{U} = \frac{1}{6}\begin{bmatrix}-2 & 0 & 0 & -1\\ 0 & 2 & 1 & 0\\ 0 & 1 & 2 & 0\\ -1 & 0 & 0 & -2\end{bmatrix} \ , \qquad \text{(A.28)}$$

$$\mathbf{N} = \frac{1}{6}\begin{bmatrix}-2 & -1 & 0 & 0\\ -1 & -2 & 0 & 0\\ 0 & 0 & 2 & 1\\ 0 & 0 & 1 & 2\end{bmatrix} \ , \qquad \text{(A.29)}$$

$$\mathbf{K} = \frac{1}{12}\begin{bmatrix}2 & 2 & 1 & 1\\ -2 & -2 & -1 & -1\\ -1 & -1 & -2 & -2\\ 1 & 1 & 2 & -2\end{bmatrix} \ , \qquad \text{(A.30)}$$

$$\mathbf{L} = \frac{1}{12}\begin{bmatrix}2 & 1 & 1 & 2\\ 1 & 2 & 2 & 1\\ -1 & -2 & -2 & -1\\ -2 & -1 & -1 & -2\end{bmatrix} \ , \qquad \text{(A.31)}$$

and

$$\mathbf{M} = \frac{1}{36}\begin{bmatrix}4 & 2 & 1 & 2\\ 2 & 4 & 2 & 1\\ 1 & 2 & 4 & 2\\ 2 & 1 & 2 & 4\end{bmatrix} \ . \qquad \text{(A.32)}$$

The BLD S_n equations are given by Eqs. (A.19) through (A.27).

APPENDIX B

THE BILINEAR-DISCONTINUOUS DIFFUSION EQUATIONS

In this appendix, we derive the BLD mixed finite element diffusion scheme for the mesh shown in Fig. 1. We derive the equations for the scalar flux at each corner ($k = 1,4$) of the interior mesh cells ($1 \le i \le I$ and $1 \le j \le J$) and the corner scalar fluxes in the ghost cells along the left boundary ($i = 0$ and $1 \le j \le J$). Analogous equations (not derived) are given for the scalar fluxes in the remainder of the void cells along the outer boundaries.

We start with the diffusion problem:

$$\frac{\partial}{\partial x} J^x(x,y) + \frac{\partial}{\partial y} J^y(x,y) + \sigma_a(x,y)\phi(x,y) = Q(x,y) \ ,$$

$$x_{1/2} < x < x_{I+1/2},\ y_{1/2} < y < y_{J+1/2} \ , \qquad \text{(B.1)}$$

where

$$J^x(x,y) = -D(x,y)\frac{\partial}{\partial x}\phi(x,y) \ , \qquad \text{(B.2)}$$

$$J^y(x,y) = -D(x,y)\frac{\partial}{\partial y}\phi(x,y) \ , \qquad \text{(B.3)}$$

and

$$D(x,y) = \frac{1}{3\sigma_t(x,y)} \ . \qquad \text{(B.4)}$$

Marshak boundary conditions are used to relate the currents and scalar fluxes on the boundaries as follows:

$$J^x(x_{1/2},y) = \frac{(\alpha_L - 1)}{(1+\alpha_L)}\frac{\phi(x_{1/2},y)}{2} \ , \qquad \text{(B.5)}$$

$$J^x(x_{I+1/2},y) = \frac{(1-\alpha_R)}{(1+\alpha_R)}\frac{\phi(x_{I+1/2},y)}{2} \ , \qquad \text{(B.6)}$$

$$J^y(x,y_{1/2}) = \frac{(\alpha_B - 1)}{(1+\alpha_B)}\frac{\phi(x,y_{1/2})}{2} \ , \qquad \text{(B.7)}$$

and

$$J^y(x,y_{J+1/2}) = \frac{(1-\alpha_T)}{(1+\alpha_T)}\frac{\phi(x,y_{J+1/2})}{2} \ , \qquad \text{(B.8)}$$

where $\alpha_{L(R)(B)(T)} = 0$ for vacuum boundaries and $\alpha_{L(R)(B)(T)} = 1$ for reflecting boundaries.

We derive the interior mesh cell equations first. To do this, we operate on Eq. (B.1) by

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} B_{k,i,j}(x,y)(\cdot)\, dx\, dy \ ,$$

where $B_{k,i,j}(x,y)$ $(k = 1,4)$ are given by Eqs. (A.7) through (A.10), to obtain

$$\begin{aligned}
&\int_{y_{j-1/2}}^{y_{j+1/2}} [B_{k,i,j}(x_{i+1/2},y)J^x(x_{i+1/2},y) \\
&\quad - B_{k,i,j}(x_{i-1/2},y)J^x(x_{i-1/2},y)]\, dy \\
&\quad + \int_{x_{i-1/2}}^{x_{i+1/2}} [B_{k,i,j}(x,y_{j+1/2})J^x(x,y_{j+1/2}) \\
&\quad - B_{k,i,j}(x,y_{j-1/2})J^x(x,y_{j-1/2})]\, dx \\
&\quad + \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \left[J^x(x,y)\frac{\partial}{\partial x} B_{k,i,j}(x,y) \right. \\
&\quad \left. + J^y(x,y)\frac{\partial}{\partial y} B_{k,i,j}(x,y)\right] dx\, dy \\
&\quad + \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} B_{k,i,j}(x,y) \\
&\quad \times [\sigma_{a,i,j}\phi(x,y) - Q(x,y)]\, dx\, dy = 0 \ , \\
&\qquad 1 \le i \le I,\ 1 \le j \le J \ . \qquad \text{(B.9)}
\end{aligned}$$

We define

$$\begin{aligned}
J^x(x_{i+1/2},y) &= \frac{\phi(x_{i+1/2}^-,y)}{4} + \frac{J^x(x_{i+1/2}^-,y)}{2} \\
&\quad - \frac{\phi(x_{i+1/2}^+,y)}{4} + \frac{J^x(x_{i+1/2}^+,y)}{2}
\end{aligned} \qquad \text{(B.10)}$$

and

$$\begin{aligned}
J^y(x,y_{j+1/2}) &= \frac{\phi(x,y_{j+1/2}^-)}{4} + \frac{J^y(x,y_{j+1/2}^-)}{2} \\
&\quad - \frac{\phi(x,y_{j+1/2}^+)}{4} + \frac{J^y(x,y_{i+1/2}^+)}{2} \ ,
\end{aligned} \qquad \text{(B.11)}$$

where

$$J^x(x_{i+1/2}^-,y) = \begin{cases} \dfrac{(\alpha_L - 1)}{(1+\alpha_L)} \dfrac{\phi(x_{i+1/2}^-,y)}{2} \ , \\ \qquad i = 0 \\ -D(x_{i+1/2}^-,y)\dfrac{\partial}{\partial x}\phi(x_{i+1/2}^-,y) \ , \\ \qquad 1 \le i \le I \ , \end{cases} \qquad \text{(B.12)}$$

$$J^x(x_{i+1/2}^+,y) = \begin{cases} -D(x_{i+1/2}^+,y)\dfrac{\partial}{\partial x}\phi(x_{i+1/2}^+,y) \ , \\ \qquad 0 \le i \le I-1 \\ \dfrac{(1 - \alpha_R)}{(1+\alpha_R)} \dfrac{\phi(x_{i+1/2}^+,y)}{2} \ , \\ \qquad i = I \ , \end{cases} \qquad \text{(B.13)}$$

$$J^y(x,y_{j+1/2}^-) = \begin{cases} \dfrac{(\alpha_B - 1)}{(1+\alpha_B)} \dfrac{\phi(x,y_{j+1/2}^-)}{2} \ , \\ \qquad j = 0 \\ -D(x,y_{j+1/2}^-)\dfrac{\partial}{\partial y}\phi(x,y_{j+1/2}^-) \ , \\ \qquad 1 \le j \le J \ , \end{cases} \qquad \text{(B.14)}$$

and

$$J^y(x,y_{i+1/2}^+) = \begin{cases} -D(x,y_{j+1/2}^+)\dfrac{\partial}{\partial y}\phi(x,y_{j+1/2}^+) \ , \\ \qquad 0 \le j \le J-1 \\ \dfrac{(1 - \alpha_R)}{(1+\alpha_R)} \dfrac{\phi(x,y_{j+1/2}^+)}{2} \ , \\ \qquad j = J \ . \end{cases} \qquad \text{(B.15)}$$

Within each cell, we make the following approximations:

$$\phi(x,y) = \sum_{k'=1}^{4} \phi_{k',i,j} B_{k',i,j}(x,y) \ , \qquad \text{(B.16)}$$

$$J^x(x,y) = -D_{i,j} \sum_{k'=1}^{4} \phi_{k',i,j} \frac{\partial}{\partial x} B_{k',i,j}(x,y) \ , \qquad \text{(B.17)}$$

$$J^y(x,y) = -D_{i,j} \sum_{k'=1}^{4} \phi_{k',i,j} \frac{\partial}{\partial y} B_{k',i,j}(x,y) \ , \qquad \text{(B.18)}$$

$$\begin{aligned}
&D(x_{i+1/2}^-,y)\frac{\partial}{\partial x}\phi(x_{i+1/2}^-,y) \\
&\quad = D_{i,j} \sum_{k'=1}^{4} \phi_{k',i,j} \frac{\partial}{\partial x} B_{k',i,j}(x_{i+1/2},y) \ ,
\end{aligned} \qquad \text{(B.19)}$$

$$D(x^+_{i+1/2},y)\frac{\partial}{\partial x}\phi(x^+_{i+1/2},y) = D_{i+1,j}\sum_{k'=1}^{4}\phi_{k',i+1,j}\frac{\partial}{\partial x}B_{k',i+1,j}(x_{i+1/2},y) \ , \tag{B.20}$$

$$D(x,y^-_{j+1/2})\frac{\partial}{\partial t}\phi(x,y^-_{j+1/2}) = D_{i,j}\sum_{k'=1}^{4}\phi_{k',i,j}\frac{\partial}{\partial t}B_{k',i,j}(x,y_{j+1/2}) \ , \tag{B.21}$$

$$D(x,y^+_{j+1/2})\frac{\partial}{\partial y}\phi(x,y^+_{j+1/2}) = D_{i,j+1}\sum_{k'=1}^{4}\phi_{k',i,j+1}\frac{\partial}{\partial y}\beta_{k',i,j+1}(x,y_{j+1/2}) \ , \tag{B.22}$$

and

$$Q(x,y) = \sum_{k'=1}^{4} q_{k',i,j}B_{k',i,j}(x,y) \ . \tag{B.23}$$

Combining Eqs. (B.9) through (B.22) and performing the integrals, we obtain the following equations:

$$\begin{aligned}
&\left[\left(\beta_1+\beta_2+\frac{\beta_7}{3}\right)\phi_1+\left(\frac{\beta_2}{2}-\beta_3+\frac{\beta_7}{6}\right)\phi_2\right]_{i,j} \\
&\quad+\left[\left(-\frac{\beta_3}{2}-\frac{\beta_4}{2}+\frac{\beta_7}{12}\right)\phi_3+\left(\frac{\beta_1}{2}-\beta_4+\frac{\beta_7}{6}\right)\phi_4\right]_{i,j} \\
&\quad-\left(\beta_3\phi_1+\beta_5\phi_2+\frac{\beta_5}{2}\phi_3+\frac{\beta_3}{2}\phi_4\right)_{i-1,j} \\
&\quad-\left(\beta_4\phi_1+\frac{\beta_4}{2}\phi_2+\frac{\beta_6}{2}\phi_3+\beta_6\phi_4\right)_{i,j-1} \\
&=\frac{\Delta x_i\Delta y_j}{36}(4q_1+2q_2+q_3+2q_4)_{i,j} \ ,
\end{aligned}\tag{B.24}$$

$$\begin{aligned}
&\left[\left(\frac{\beta_2}{2}-\beta_3+\frac{\beta_7}{6}\right)\phi_1+\left(\beta_1+\beta_2+\frac{\beta_7}{3}\right)\phi_2\right]_{i,j} \\
&\quad+\left[\left(\frac{\beta_1}{2}-\beta_4+\frac{\beta_7}{6}\right)\phi_3+\left(-\frac{\beta_3}{2}-\frac{\beta_4}{2}+\frac{\beta_7}{12}\right)\phi_4\right]_{i,j} \\
&\quad-\left(\beta_5\phi_1+\beta_3\phi_2+\frac{\beta_3}{2}\phi_3+\frac{\beta_5}{2}\phi_4\right)_{i+1,j} \\
&\quad-\left(\frac{\beta_4}{2}\phi_1+\beta_4\phi_2+\beta_6\phi_3+\frac{\beta_6}{2}\phi_4\right)_{i,j-1} \\
&=\frac{\Delta x_i\Delta y_j}{36}(2q_1+4q_2+2q_3+q_4)_{i,j} \ ,
\end{aligned}\tag{B.25}$$

$$\begin{aligned}
&\left[\left(-\frac{\beta_3}{2}-\frac{\beta_4}{2}+\frac{\beta_7}{12}\right)\phi_1+\left(\frac{\beta_1}{2}-\beta_4+\frac{\beta_7}{6}\right)\phi_2\right]_{i,j} \\
&\quad+\left[\left(\beta_1+\beta_2+\frac{\beta_7}{3}\right)\phi_3+\left(\frac{\beta_2}{2}-\beta_3+\frac{\beta_7}{6}\right)\phi_4\right]_{i,j} \\
&\quad-\left(\frac{\beta_6}{2}\phi_1+\beta_6\phi_2+\beta_4\phi_3+\frac{\beta_4}{2}\phi_4\right)_{i,j+1} \\
&\quad-\left(\frac{\beta_5}{2}\phi_1+\frac{\beta_3}{2}\phi_2+\beta_3\phi_3+\beta_5\phi_4\right)_{i+1,j} \\
&=\frac{\Delta x_i\Delta y_j}{36}(q_1+2q_2+4q_3+2q_4)_{i,j} \ ,
\end{aligned}\tag{B.26}$$

and

$$\begin{aligned}
&\left[\left(\frac{\beta_1}{2}-\beta_4+\frac{\beta_7}{6}\right)\phi_1+\left(-\frac{\beta_3}{2}-\frac{\beta_4}{2}+\frac{\beta_7}{12}\right)\phi_2\right]_{i,j} \\
&\quad+\left[\left(\frac{\beta_2}{2}-\beta_3+\frac{\beta_7}{6}\right)\phi_3+\left(\beta_1+\beta_2+\frac{\beta_7}{3}\right)\phi_4\right]_{i,j} \\
&\quad-\left(\beta_6\phi_1+\frac{\beta_6}{2}\phi_2+\frac{\beta_4}{2}\phi_3+\beta_4\phi_4\right)_{i,j+1} \\
&\quad-\left(\frac{\beta_3}{2}\phi_1+\frac{\beta_5}{2}\phi_2+\beta_5\phi_3+\beta_3\phi_4\right)_{i-1,j} \\
&=\frac{\Delta x_i\Delta y_j}{36}(2q_1+q_2+2q_3+4q_4)_{i,j} \ ,
\end{aligned}\tag{B.27}$$

where

$$\beta_{1,i,j}=\left(\frac{1}{4}+\frac{D_{i,j}}{2\Delta x_i}\right)\frac{\Delta y_j}{3} \ , \quad 1\le i\le I,\ 1\le j\le J \ , \tag{B.28}$$

$$\beta_{2,i,j}=\left(\frac{1}{4}+\frac{D_{i,j}}{2\Delta y_j}\right)\frac{\Delta x_i}{3} \ , \quad 1\le i\le I,\ 1\le j\le J \ , \tag{B.29}$$

$$\beta_{3,i,j}=\begin{cases}0 \ , & i=0 \ , \\ \dfrac{D_{i,j}\Delta y_j}{6\Delta x_i} \ , & 1\le i\le I,\ 1\le j\le J \ , \\ 0 \ , & i=I+1 \ , \end{cases}\tag{B.30}$$

$$\beta_{4,i,j}=\begin{cases}0 \ , & j=0 \ , \\ \dfrac{D_{i,j}\Delta x_i}{6\Delta y_j} \ , & 1\le j\le J,\ 1\le j\le J \ , \\ 0 \ , & j=J+1 \ , \end{cases}\tag{B.31}$$

$$\beta_{5,i,j} = \begin{cases} \dfrac{\alpha_L \Delta y_j}{6(1+\alpha_L)} , & i = 0 , \\ \left(\dfrac{1}{4} - \dfrac{D_{i,j}}{2\Delta x_i}\right)\dfrac{\Delta y_j}{3} , & \\ & 1 \le i \le I,\ 1 \le i \le I , \\ \dfrac{\alpha_R \Delta y_j}{6(1+\alpha_R)} , & i = I+1 , \end{cases} \tag{B.32}$$

$$\beta_{6,i,j} = \begin{cases} \dfrac{\alpha_B \Delta y_j}{6(1+\alpha_B)} , & j = 0 , \\ \left(\dfrac{1}{4} - \dfrac{D_{i,j}}{2\Delta y_j}\right)\dfrac{\Delta x_i}{3} , & \\ & 1 \le j \le J,\ 1 \le i \le I , \\ \dfrac{\alpha_T \Delta y_j}{6(1+\alpha_T)} , & j = J+1 , \end{cases} \tag{B.33}$$

and

$$\beta_{7,i,j} = \frac{\sigma_{a,i,j}\Delta x_i \Delta y_j}{3} , \quad 1 \le i \le I,\ 1 \le j \le J . \tag{B.34}$$

Now, we derive the equations for the void cells at $i = 0$ $(1 \le j \le J)$, where

$$\frac{\partial}{\partial x} J^x(x,y) = 0 , \quad x < x_{1/2}, y_{1/2} \le y \le y_{j+1/2} . \tag{B.35}$$

We define the basis functions as

$$B_{1,0,j}(x,y) = B_{1,0,j}(y) = 0 , \tag{B.36}$$

$$B_{2,0,j}(x,y) = B_{2,0,j}(y) = \left(1 - \frac{y - y_{j-1/2}}{\Delta y_j}\right) , \tag{B.37}$$

$$B_{3,0,j}(x,y) = B_{3,0,j}(y) = \left(\frac{y - y_{j-1/2}}{\Delta y_j}\right) , \tag{B.38}$$

and

$$B_{4,0,j}(x,y) = B_{4,0,j}(y) = 0 . \tag{B.39}$$

We operate on Eq. (B.35) by

$$\int_{x_{-1/2}}^{x_{1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} B_{k,0,j}(x,y)(\cdot)\, dx\, dy$$

and obtain

$$\int_{y_{j-1/2}}^{y_{j+1/2}} B_{k,0,j}(y)\,[J^x(x_{1/2},y) - J^x(x_{-1/2},y)]\, dx = 0 , \tag{B.40}$$

where $J^x(x_{1/2})$ is given by Eq. (B.10) and Eqs. (B.12) and (B.13) evaluated at $i = 0$. We then define

$$J^x(x_{-1/2}) = J^x(x_{1/2}^-) . \tag{B.41}$$

After performing the integrals, we obtain

$$\frac{1}{\alpha_L}\left(\beta_5\phi_2 + \frac{\beta_5}{2}\phi_3\right)_{0,j} - \left(\beta_5\phi_1 + \beta_3\phi_2 + \frac{\beta_3}{2}\phi_3 + \frac{\beta_5}{2}\phi_4\right)_{1,j} = 0 \tag{B.42}$$

and

$$\frac{1}{\alpha_L}\left(\frac{\beta_5}{2}\phi_2 + \beta_5\phi_3\right)_{0,j} - \left(\frac{\beta_5}{2}\phi_1 + \frac{\beta_3}{2}\phi_2 + \beta_3\phi_3 + \beta_5\phi_4\right)_{1,j} = 0 . \tag{B.43}$$

Analogous derivations can be made for the other void cells to give

$$\frac{1}{\alpha_R}\left(\beta_5\phi_1 + \frac{\beta_5}{2}\phi_4\right)_{I+1,j} - \left(\beta_3\phi_1 + \beta_5\phi_2 + \frac{\beta_5}{2}\phi_3 + \frac{\beta_3}{2}\phi_4\right)_{I,j} = 0 , \tag{B.44}$$

$$\frac{1}{\alpha_R}\left(\frac{\beta_5}{2}\phi_1 + \beta_5\phi_4\right)_{I+1,j} - \left(\frac{\beta_3}{2}\phi_1 + \frac{\beta_5}{2}\phi_2 + \beta_5\phi_3 + \beta_3\phi_4\right)_{I,j} = 0 , \tag{B.45}$$

$$\frac{1}{\alpha_B}\left(\beta_6\phi_3 + \frac{\beta_6}{2}\phi_4\right)_{i,0} - \left(\frac{\beta_6}{2}\phi_1 + \beta_6\phi_2 + \beta_4\phi_3 + \frac{\beta_4}{2}\phi_4\right)_{i,1} = 0 , \tag{B.46}$$

$$\frac{1}{\alpha_B}\left(\frac{\beta_6}{2}\phi_3 + \beta_6\phi_4\right)_{i,0} - \left(\beta_6\phi_1 + \frac{\beta_6}{2}\phi_2 + \frac{\beta_4}{2}\phi_3 + \beta_4\phi_4\right)_{i,1} = 0 , \tag{B.47}$$

$$\frac{1}{\alpha_T}\left(\beta_6\phi_1 + \frac{\beta_6}{2}\phi_2\right)_{i,J+1} - \left(\beta_4\phi_1 + \frac{\beta_4}{2}\phi_2 + \frac{\beta_6}{2}\phi_3 + \beta_6\phi_4\right)_{i,J} = 0 , \tag{B.48}$$

and

$$\frac{1}{\alpha_T}\left(\frac{\beta_6}{2}\phi_1+\beta_6\phi_2\right)_{i,J+1} - \left(\frac{\beta_4}{2}\phi_1+\beta_4\phi_2+\beta_6\phi_3+\frac{\beta_6}{2}\phi_4\right)_{i,J} = 0 \ . \tag{B.49}$$

The BLD mixed finite element diffusion equations are given by Eqs. (B.24) through (B.27) and Eqs. (B.42) through (B.49).

In our acceleration scheme, these equations are used to calculate approximate errors for the S_n BLD scalar flux iterates. For this application, the BLD diffusion source vector is equal to the residual vector associated with the BLD S_n iterate. We can write the BLD S_n residual vector elements for cell i,j as follows:

$$R_{k,i,j} = \sigma_{s,i,j}(\phi_{j,k}^{new} - \phi_{j,k}^{old}) \ , \quad k = 1,4, \ i = 1,I, \ j = 1,J \ , \tag{B.50}$$

where ϕ^{new} denotes the scalar flux after the latest S_n sweep and ϕ^{old} denotes the flux at the beginning of the latest sweep. Since there are no void cells associated with the S_n BLD equations, there are no distributed source terms in the BLD diffusion void cells.

The principle to be followed in defining the boundary conditions for the BLD diffusion equations is universal for any two-level or two-grid system. The boundary conditions for the coarse-grid equations are chosen so that the accelerated fine-grid solution (i.e., the sum of the unaccelerated fine-grid iterate and the coarse-grid estimate of the fine-grid error) satisfies (at least approximately) the prescribed boundary conditions for the fine-grid solution. This principle is discussed in detail for the case of S_2-synthetic acceleration in Ref. 7. Assuming that the S_n iterate after each transport sweep exactly meets the prescribed boundary conditions, this principle leads to the following BLD diffusion boundary conditions:

1. If the S_n boundary condition is either vacuum or reflective, the BLD diffusion boundary condition is identical to the S_n boundary condition.

2. If the S_n boundary condition is a source condition, the BLD diffusion boundary condition is vacuum.

If the S_n iterate after each transport sweep does not meet the prescribed boundary conditions, we must define boundary residuals for the S_n equations that become boundary sources in the BLD diffusion equations. For instance, suppose that a reflective condition is prescribed for the S_n equations at a boundary; then, the following equation is supposed to be satisfied at that boundary:

$$j^{in} = j^{out} \ , \tag{B.51}$$

where j^{in} and j^{out} denote the incoming and outgoing half-range currents, respectively:

$$j^{in} = \int_{\mathbf{\Omega}\cdot\boldsymbol{n}>0} \psi(\mathbf{\Omega})\mathbf{\Omega}\cdot\boldsymbol{n}\, d\Omega \tag{B.52}$$

and

$$j^{out} = \int_{\mathbf{\Omega}\cdot\boldsymbol{n}<0} \psi(\mathbf{\Omega})|\mathbf{\Omega}\cdot\boldsymbol{n}|\, d\Omega \ , \tag{B.53}$$

and where $\boldsymbol{n}$ denotes the inward-directed normal surface vector. In accordance with Eq. (6), the boundary residual associated with Eq. (B.51) is defined as follows:

$$r^{in} = j^{out} - j^{in} \ , \tag{B.54}$$

where the half-range currents are evaluated after the transport sweep. Thus, the equation actually satisfied after the transport sweep is not Eq. (B.51), but rather

$$j^{in} = j^{out} - r^{in} \ . \tag{B.55}$$

Note that r^{in} has units of incoming current. This boundary residual becomes a boundary source in the BLD diffusion equations. Specifically, the diffusion solution at the boundary satisfies a reflective condition with a source:

$$\delta j^{in} = \delta j^{out} + r^{in} \ , \tag{B.56}$$

where δj denotes the BLD diffusion half-range current. Adding Eqs. (B.55) and (B.56), we find that the accelerated S_n iterate (i.e., the sum of the unaccelerated S_n iterate and the diffusion error estimate) satisfies the prescribed reflective boundary condition in a Marshak sense:

$$j^{in} + \delta j^{in} = j^{out} + \delta j^{out} \ . \tag{B.57}$$

In general, if one defines S_n boundary residuals in analogy with Eq. (B.54) and applies the principle that the accelerated iterate must satisfy the prescribed boundary condition in a Marshak sense, one obtains the following boundary conditions for the BLD diffusion equations:

1. If the S_n boundary condition is either vacuum or source, the BLD diffusion boundary condition is a source condition with the diffusion boundary source equal to the S_n boundary residual.

2. If the S_n boundary condition is a reflective condition, the BLD diffusion boundary condition is reflective with a boundary source equal to the S_n boundary residual, e.g., Eq. (B.56).

Boundary sources are easily added to Eqs. (B.5) through (B.8) via the Marshak condition. For instance, Eq. (B.5) becomes

$$J^x(x_{1/2},y) = \frac{(\alpha_L - 1)}{(1+\alpha_L)}\frac{\phi(x_{1/2},y)}{2} + \frac{2j^L(x_{1/2},y)}{(1+\alpha_L)} \ , \tag{B.58}$$

where j^L denotes the incoming half-range current incident along the left boundary.

APPENDIX C

THE BILINEAR-CONTINUOUS DIFFUSION SCHEME

In this appendix, we derive the BLC diffusion scheme, which is derived directly from the BLD diffusion equations. We first derive the BLC equation for the scalar flux at an interior cell vertex: $\Phi_{i+1/2,j+1/2}$, where $0 \le i \le I$ and $0 \le j \le J$. We then derive the scalar flux on the left boundary $\Phi_{1/2,j+1/2}$, where $(1 \le j \le J-1)$. Finally, we derive the scalar flux at the bottom-left corner of the problem $\Phi_{1/2,1/2}$. Analogous equations (not given) can be derived for the right, bottom, and top boundaries as well as the bottom-right, top-left, and top-right corners of the problem.

For the interior cell vertices, we add Eq. (B.24) evaluated at the i,j'th cell, Eq. (B.25) evaluated at the $i-1,j$'th cell, Eq. (B.26) evaluated at the $i-1,j-1$'th cell, and Eq. (B.27) evaluated at the $i,j-1$'th cell. We then assume that

$$\begin{aligned} \phi_{1,i,j} &= \phi_{2,i-1,j} = \phi_{3,i-1,j-1} = \phi_{4,i,j-1} \\ &\equiv \Phi_{i+1/2,j+1/2} \ . \end{aligned} \tag{C.1}$$

This gives

$$\begin{aligned} &\{2[(\beta_3+\beta_4)_{i,j} + (\beta_3+\beta_4)_{i-1,j} \\ &\quad + (\beta_3+\beta_4)_{i,j-1} + (\beta_3+\beta_4)_{i-1,j-1}] \\ &\quad + \tfrac{1}{3}(\beta_{7,i,j} + \beta_{7,i-1,j} + \beta_{7,i,j-1} + \beta_{7,i-1,j-1})\} \\ &\quad \times \Phi_{i-1/2,j-1/2} \\ &\{-2[(\beta_3+\beta_4)_{i-1,j} + (\beta_3+\beta_4)_{i-1,j-1}] \\ &\quad + \tfrac{1}{3}(\beta_{7,i-1,j} + \beta_{7,i-1,j-1})\}\Phi_{i-3/2,j-1/2} \\ &\{-2[(\beta_3+\beta_4)_{i,j} + (\beta_3+\beta_4)_{i,j-1}] \\ &\quad + \tfrac{1}{3}(\beta_{7,i,j} + \beta_{7,i,j-1})\}\Phi_{i+1/2,j-1/2} \\ &\{-2[(\beta_3+\beta_4)_{i,j-1} + (\beta_3+\beta_4)_{i-1,j-1}] \\ &\quad + \tfrac{1}{3}(\beta_{7,i,j-1} + \beta_{7,i-1,j-1})\}\Phi_{i-1/2,j-3/2} \\ &\{-2[(\beta_3+\beta_4)_{i,j} + (\beta_3+\beta_4)_{i-1,j}] \\ &\quad + \tfrac{1}{3}(\beta_{7,i,j} + \beta_{7,i-1,j})\}\Phi_{i-1/2,j+1/2} \\ &(-\beta_3 - \beta_4 + \tfrac{1}{12}\beta_7)_{i-1,j-1}\Phi_{i-3/2,j-3/2} \\ &\quad + (-\beta_3 - \beta_4 + \tfrac{1}{12}\beta_7)_{i-1,j}\Phi_{i-3/2,j+1/2} \\ &(-\beta_3 - \beta_4 + \tfrac{1}{12}\beta_7)_{i,j-1}\Phi_{i+1/2,j-3/2} \\ &\quad + (-\beta_3 - \beta_4 + \tfrac{1}{12}\beta_7)_{i,j}\Phi_{i+1/2,j+1/2} \\ &\ = q_{1,i,j} + q_{2,i-1,j} + q_{3,i-1,j-1} + q_{4,i,j-1} \ , \\ &\qquad\qquad 2 \le i \le I,\ 2 \le j \le J \ . \end{aligned} \tag{C.2}$$

For the left boundary, we add Eq. (B.24) evaluated at the $1,j$'th cell, Eq. (B.42) evaluated at the $0,j$'th cell, Eq. (B.43) evaluated at the $0,j-1$'th cell and Eq. (B.27) evaluated at the $1,j-1$'th cell. This gives

$$\begin{aligned} &[(2\beta_3 + 2\beta_4 + \tfrac{1}{3}\beta_7)_{1,j} + (2\beta_3 + 2\beta_4 + \tfrac{1}{3}\beta_7)_{1,j-1} \\ &\quad + \beta_{5,0,j} + \beta_{5,0,j-1}]\Phi_{3/2,j-1/2} \\ &[(-2\beta_3 + 2\beta_4 + \tfrac{1}{6}\beta_7)_{1,j} \\ &\quad + (-2\beta_3 + 2\beta_4 + \tfrac{1}{6}\beta_7)_{1,j-1}]\Phi_{1/2,j-1/2} \\ &[(2\beta_3 - 2\beta_4 + \tfrac{1}{6}\beta_7)_{1,j-1} + \tfrac{1}{2}\beta_{5,0,j-1}]\Phi_{1/2,j-3/2} \\ &\quad + [(2\beta_3 - 2\beta_4 + \tfrac{1}{6}\beta_7)_{1,j} + \tfrac{1}{2}\beta_{5,0,j}]\Phi_{1/2,j+1/2} \\ &(-\beta_3 - \beta_4 + \tfrac{1}{12}\beta_7)_{1,j-1}\Phi_{3/2,j-3/2} \\ &\quad + (-\beta_3 - \beta_4 + \tfrac{1}{12}\beta_7)_{1,j}\Phi_{3/2,j+1/2} \\ &\ = q_{1,1,j} + q_{2,0,j} + q_{3,0,j-1} + q_{4,1,j-1} \ , \\ &\qquad\qquad 2 \le j \le J \ . \end{aligned} \tag{C.3}$$

For the bottom-left corner of the mesh, we add Eq. (B.24) evaluated at the 1,1'th cell, Eq. (B.42) evaluated at the 0,1'th cell and Eq. (B.47) evaluated at the 1,0 cell. This gives

$$\begin{aligned} &[(2\beta_3 + 2\beta_4 + \tfrac{1}{3}\beta_7)_{1,1} + \beta_{5,0,1} + \beta_{6,1,0}]\Phi_{1/2,1/2} \\ &[(-2\beta_3 + 2\beta_4 + \tfrac{1}{6}\beta_7)_{1,1} + \beta_{6,1,0}]\Phi_{3/2,1/2} \\ &\quad + [(2\beta_3 - 2\beta_4 + \tfrac{1}{6}\beta_7)_{1,1} + \beta_{5,0,1}]\Phi_{1/2,3/2} \\ &\quad + (-\beta_3 - \beta_4 + \tfrac{1}{12}\beta_7)_{1,1}\Phi_{3/2,3/2} \\ &\ = q_{1,1,1} + q_{2,0,1} + q_{4,1,0} \ . \end{aligned} \tag{C.4}$$

In our acceleration scheme, these equations are used to calculate the approximate errors associated with the BLD diffusion iterates. For this application, the BLC diffusion source vector is equal to the projection from the BLD diffusion grid of the BLD diffusion residual vector onto the BLC diffusion grid. As previously discussed, this projection is performed simply by summing the four BLD residuals associated with each vertex to obtain a single vertex residual. There is no simplified expression for the BLD diffusion residuals. One must simply calculate the residual vector according to Eq. (6). Once the BLC diffusion solution has been obtained, the error estimates are interpolated onto the BLD diffusion grid by assigning the single BLC error associated with each vertex to each of the four BLD fluxes associated with that vertex.

The boundary conditions for the BLC equations are automatically obtained if one derives the BLC equations directly from the BLD diffusion equations by assuming continuity of the BLD solution and then summing the four BLD equations associated with each vertex. In general, one finds that the BLC diffusion boundary conditions are identical to the BLD diffusion boundary conditions except that the BLC conditions may also include a boundary source. These boundary sources automatically arise from the BLD diffusion residuals associated with the equations for the void fluxes. Thus, a special calculation of boundary residuals is not required for the BLD diffusion equations as it is for the S_n equations.

ACKNOWLEDGMENT

This work was performed under the auspices of the U.S. Department of Energy.

REFERENCES

1. M. L. ADAMS and W. R. MARTIN, "Diffusion Synthetic Acceleration of Discontinuous Finite Element Transport Iterations," *Nucl. Sci. Eng.*, **111**, 145 (1992).

2. T. A. WAREING, E. W. LARSEN, and M. L. ADAMS, "Diffusion Accelerated Discontinuous Finite Element Schemes for the S_n Equations in Slab and X,Y Geometries," *Proc. Int. Topl. Mtg. Advances in Mathematics, Computations, and Reactor Physics*, April 28-May 2, 1991, Pittsburgh, Pennsylvania, Vol. 3, Sec. 11.1, p. 2-1, American Nuclear Society (1991).

3. J. E. DENDY, Jr., "Black-Box Multigrid," *J. Comput. Phys.*, **48**, 366 (1982).

4. L. A. HAGEMAN and D. M. YOUNG, *Applied Iterative Methods*, Academic Press, New York (1981).

5. T. A. WAREING, W. F. WALTERS, and J. E. MOREL, "Diffusion-Synthetic Acceleration of the 2-D *X-Y* S_n Equations with Linear-Bilinear Nodal Differencing," to be submitted to *Nucl. Sci. Eng.*

6. J. E. MOREL, J. E. DENDY, M. L. HALL, and S. W. WHITE, "A Cell-Centered Lagrangian-Mesh Diffusion Differencing Scheme," *J. Comput. Phys.*, **103**, 286 (1992).

7. L. J. LORENCE, Jr., J. E. MOREL, and E. W. LARSEN, "An S_2 Synthetic Acceleration Scheme for the One-Dimensional S_n Equations with Linear Discontinuous Spatial Differencing," *Nucl. Sci. Eng.*, **101**, 341 (1992).