A Local Support-Operators Diffusion Discretization Scheme for Hexahedral Meshes

J. E. Morel, Michael L. Hall, and Mikhail J. Shashkov

University of California, Los Alamos National Laboratory, Los Alamos, New Mexico 87545
E-mail: jim@lanl.gov; hall@lanl.gov; misha@t7.lanl.gov

We derive a cell-centered 3-D diffusion differencing scheme for unstructured hexahedral meshes using the local support-operators method. Our method is said to be local because it yields a sparse matrix representation for the diffusion equation, whereas the traditional support-operators method yields a dense matrix representation. The diffusion discretization scheme that we have developed offers several advantages relative to existing schemes. Most importantly, it offers second-order accuracy on reasonably well-behaved nonsmooth meshes, rigorously treats material discontinuities, and has a symmetric positive-definite coefficient matrix. The order of accuracy is demonstrated computationally rather than theoretically. Rigorous treatment of material discontinuities implies that the normal component of the flux is continuous across such discontinuities while the parallel components may be either continuous or discontinuous in accordance with the exact solution to the problem being considered. The only disadvantage of the method is that it has both cell-centered and face-centered scalar unknowns as opposed to just cell-center scalar unknowns. Computational examples are given which demonstrate the accuracy and cost of the new scheme. © 2001 Academic Press

Key Words: diffusion; support-operators method; unstructured mesh.

1. INTRODUCTION

The purpose of this paper is to present a local support-operators diffusion discretization for unstructured 3-D hexahedral meshes. We use the standard finite-element definition for hexahedra [1]. The method that we present is a generalization of a similar scheme for 2-D $r - z$ quadrilateral meshes that was developed by Morel, Roberts, and Shashkov [2]. Our focus is the discretization of the diffusion operator rather than any particular type of diffusion equation. For demonstration purposes, we choose to solve a linear diffusion equation of the

338

form

$$-\vec{\nabla} \cdot D\vec{\nabla}\phi = Q, \tag{1}$$

where ϕ denotes a scalar function that we refer to as the intensity, D denotes the diffusion coefficient, and Q denotes the source or driving function. It is sometimes useful to express Eq. (1) in terms of a vector function, $\vec{F}$, that we refer to as the flux:

$$\vec{F} = -D\vec{\nabla}\phi. \tag{2}$$

We have taken the terms "intensity" and "flux" from the radiative transfer literature [3], but we have not explicitly considered the radiative diffusion equation because the subject of this paper relates only to the discretization of the diffusion operator. Our discretization can be used in any type of diffusion calculation, e.g., time-dependent, steady-state, linear, or nonlinear.

We define a cell-centered diffusion discretization scheme as one that numerically preserves the integral of Eq. (1) over each spatial cell. In particular, substituting from Eq. (2) into Eq. (1) and integrating that equation over a cell volume, we obtain

$$\oint_{\partial V} \vec{F} \cdot \vec{n} \, dA = \int_V Q \, dV, \tag{3}$$

where ∂V denotes the cell surface, $\vec{n}$ denotes the outward-directed unit surface normal, and V denotes the cell volume. Note that we used the divergence theorem to convert the first integral in Eq. (3) from a volume integral to a surface integral. In physical terms, Eq. (3) generally represents a statement of particle or energy conservation over the cell. Thus, we can simply state that cell-centered schemes (as we define them) are conservative over each mesh cell.

If one considers only nonorthogonal meshes with material discontinuities, existing vertex-centered diffusion discretizations are generally more advanced than cell-centered discretizations. This is primarily so because of the enormous success of Galerkin finite-element methods [1] and variants of those methods. Nonetheless, there are applications for which cell-centered schemes appear to yield superior accuracy relative to vertex-centered schemes. For instance, when coupling radiation diffusion calculations with cell-centered hydrodynamics calculations, a cell-centered diffusion scheme is highly desirable because it avoids certain difficulties associated with mapping between vertex-centered and cell-centered material temperatures [4].

The discretization scheme that we have developed is cell-centered, but it has intensity unknowns at both cell centers and face centers. It can be applied on unstructured hexahedral meshes. It yields second-order accurate solutions for the intensities on both smooth and nonsmooth meshes even when material discontinuities are present, and it generates a sparse symmetric positive-definite coefficient matrix. Second-order convergence has been demonstrated computationally on nonsmooth meshes, but it has not been theoretically proven.

The literature relating to cell-centered diffusion discretization schemes for nonorthogonal hexahedra is not particularly extensive. One of the earliest relevant papers appeared about 10 years ago. In particular, Rose developed a cell-centered hexahedral-mesh discretization scheme for the Laplacian operator [5]. The diffusion operator that we consider degenerates to the Laplacian operator when the diffusion coefficient is *identically one*. Unlike our scheme,

which has only the normal component of the current on each cell face, Rose's scheme has three components of the flux on each cell. Furthermore, the flux is continuous across each cell face in Rose's scheme, whereas only the normal component of the flux is continuous in our scheme. A central aspect of Rose's method is the preservation of an integral expression that is referred to as an energy principle. Our method is actually based upon the preservation of an integral identity. The energy principle used by Rose is not the same as the integral identity that we use, but they are related. In particular, the principle used by Rose can be derived from the diffusion equation together with the integral identity that we use. Rose presented a proof that his hexahedral-mesh method converges with second-order accuracy, but he provided computational results only for a $1 - D$ version of his method. Arbogast *et al.* [6] have recently developed a cell-centered expanded mixed finite-element method for solving the tensor diffusion equation on general meshes (including hexahedral meshes). Their method has only cell-center intensity unknowns if both the mesh and the diffusion tensor are smooth, but additional face-center intensities are required wherever the mesh or the diffusion tensor is nonsmooth. The coefficient matrix generated by their method is always symmetric positive definite (SPD). The method of Arbogast *et al.* actually shares some of the best properties of the standard mixed finite-element method and the hybrid mixed finite-element method. Standard mixed finite-element diffusion methods have only cell-center intenstities, but this is achieved at the cost of solving a computationally expensive saddle-point linear system. The saddle-point system can be avoided by using the hybrid mixed finite-element approach, which generates a symmetric positive-definite coefficient matrix at the expense of additional face-center unknowns. The method of Arbogast *et al.* yields an SPD coefficient matrix similar to the mixed hybrid method but it can sometimes require far fewer unknowns. Although they proved several convergence theorems for their hexahedral-mesh method, they had to assume certain mesh smoothness properties. Furthermore, Arbogast *et al.* provided computational results only for a 2-D version of their method.

Our local support-operators method is similar to hybrid mixed finite-element methods in that it is cell-centered, it has both cell-center and cell-face intensities, and it produces a coefficient matrix that is symmetric positive-definite. However, our scheme is fundamentally a finite-volume technique since basis functions never appear in our formalism. Nonetheless, a strong connection does exist between our method and hybrid mixed finite-element methods. This connection arises from the fact that the integral identity that is the basis of the support-operators method is in fact a weak form of Eq. (2). Hybrid mixed finite-element methods satisfy a weak form of Eq. (2) on specific finite-dimensional function spaces, whereas the support-operators solution satisfies a weak form of Eq. (2) in a purely discrete sense. The global support-operators method has recently been reformulated to include the use of vector basis functions on general quadrilateral meshes [7]. The basis-function version of the method recovers the finite-difference version when exact integration is replaced with certain approximate quadratures. The basis-function formulation appears to be about three times more accurate than the finite-difference formulation but both formulations exhibit the same order of convergence. Because of the complexity of the 2-D vector basis functions, the authors of [7] conclude that the improvement in accuracy does not justify the added complexity of the basis-function support-operators method. We feel that our local support-operators method for general hexahedral meshes is much simpler than hybrid mixed finite-element methods precisely because the vector basis functions for hexahedral meshes are extremely complicated [6]. More importantly, our local support-operators method converges on nonsmooth hexahedral meshes, but we have not been able to

identify any hybrid mixed finite-element methods that have been shown to converage on such meshes.

To summarize, the following combination of characteristics appear to be unique to our support-operators diffusion discretization scheme:

- It is a cell-centered discretization for unstructured hexahedral meshes.
- It has been computationally demonstrated that the scheme gives second-order convergence of the intensity on both smooth and nonsmooth meshes both with and without material discontinuities.
- It generates a sparse SPD coefficient matrix.
- It is equivalent to the standard 7-point cell-center diffusion discretization scheme [8] when the mesh is orthogonal.

We stress that some of the latest hybrid finite-element methods for hexahedral meshes require a certain degree of mesh smoothness for convergence [6], whereas our method converges on nonsmooth grids. Thus, our method clearly represents a valuable alternative to hybrid mixed finite-element methods.

The remainder of this paper is organized as follows. We first explain the central theme of our local support-operators method, and apply it to a nonorthogonal hexahedral mesh in Cartesian geometry. We next describe an approximate version of our scheme that we use as a preconditioner in conjunction with a conjugate-gradient solution technique [9]. Finally, computational results are given, followed by a summary and recommendations for future work.

2. THE SUPPORT-OPERATORS METHOD

In this section we describe the support-operators method. It is convenient at this point to define a flux operator given by $-D\vec{\nabla}$. The diffusion operator of interest is given by the product of the divergence operator and the flux operator: $-\vec{\nabla} \cdot D\vec{\nabla}$. The support-operators method is based upon the following three facts:

- Given appropriately defined scalar and vector inner products, the divergence and flux operators are adjoint to one another.
- The adjoint of an operator varies with the definition of its associated inner products, but is unique for fixed inner products.
- The product of an operator and its adjoint is a self-adjoint positive-definite operator.

The mathematical details relating to these facts are given in [10]. As explained in [10], the adjoint relationship between the flux and divergence operators is embodied in the integral identity

$$\oint_{\partial V} \phi \vec{H} \cdot \vec{n} \, dA - \int_V D^{-1}\vec{H} \cdot D\vec{\nabla}\phi \, dV = \int_V \phi \vec{\nabla} \cdot \vec{H} \, dV, \qquad (4)$$

where ϕ is an arbitrary scalar function, $\vec{H}$ is an arbitrary vector function, V denotes a volume, ∂V denotes its surface, and $\vec{n}$ denotes the outward-directed unit normal associated with that surface. This identity can be derived from the differential identity [11],

$$\vec{\nabla} \cdot (\phi \vec{H}) = \phi \vec{\nabla} \cdot \vec{H} + \vec{H} \cdot \vec{\nabla}\phi, \qquad (5)$$

and the divergence theorem [11],

$$\int_V \vec{\nabla} \cdot \vec{H} \, dV = \int_{\partial V} \vec{H} \cdot \vec{n} \, dA. \tag{6}$$

Our support-operators method can be conceptually described in the simplest terms as follows:

1. Define discrete scalar and vector spaces to be used in a discretization of Eq. (4).

2. Fully discretize all but the flux operator in Eq. (4) over a single arbitrary cell. The flux operator is left in the general form of a discrete vector as defined in Step 1.

3. Solve for the discrete flux operator (i.e., for its vector components) on a single arbitrary cell by requiring that the discrete version of Eq. (4) hold for all elements of the vector space defined in Step 1.

4. Combine the flux operator with the balance equation to obtain a discretization of Eq. (1) on a single mesh cell. This provides an equation for each cell-center intensity.

5. Connect adjacent mesh cells in such a way as to ensure that Eq. (4) is satisfied over the whole grid. This simply amounts to enforcing continuity of intensity and continuity of the normal flux component at the cell interfaces. Because each cell face on the mesh interior is shared by two cells, there are initially two distinct intensities at the center of each of these faces. The continuity of intensity condition reduces each such pair of intensities to a single intensity. The continuity of flux condition provides an equation for each of the face-center intensities on the mesh interior.

6. Use the analytic boundary conditions to obtain an expression for a "boundary normal flux component" for each cell face on the mesh boundary. Equate this "boundary normal flux component" to the normal flux component obtained via the flux operator on each boundary cell face. This provides an equation for each face-center intensity on the outer mesh boundary, and completes the specification of the diffusion matrix.

To make this process concrete, we next generate the diffusion matrix for a hexahedral mesh in Cartesian geometry. To simplify the presentation, we assume a logically rectangular mesh. However, we stress that our discretization scheme can be used with unstructured meshes as well. The assumption of a logically rectangular mesh merely simplifies our notation and mesh indexing. Our first step is to define that indexing. For reasons explained later, both global and local indices are used. Let us first consider the global indices. The cell centers carry integral global indices, e.g., (i, j, k); cell vertices carry half-integral global indices, e.g., $(i + \frac{1}{2}, j + \frac{1}{2}, k + \frac{1}{2})$; and face centers carry mixed global indices composed of both integral and half-integral indices, e.g., $(i + \frac{1}{2}, j, k)$. The global indices for four of the vertices associated with cell (i, j, k) are illustrated in Fig. 1.

Local indices allow us to uniquely define certain quantities that are associated with a vertex or face center *and* a cell. For instance, the local indices for the six faces associated with each cell are given by L, R, B, T, D, and U, which denote Left, Right, Bottom, Top, Down, and Up respectively. This local face indexing is illustrated for cell (i, j, k) in Fig. 2 and Fig. 3 together with a mapping between the local indices and the corresponding global indices. Note that the index i increases when moving from Left to Right, the index j increases when moving from Bottom to Top, and the index k increases when moving from the Down to Up. The local indices for the vertices follow directly from the face indices in that each vertex is uniquely shared by three faces of the cell. Thus, the vertex shared by the Right, Top, and Up faces is denoted by the index RTU. This vertex is illustrated in Fig. 4.

1 - (i-1/2 , j-1/ 2, k-1/2)
2 - (i+1/2 , j-1/2, k-1/2)
3 - (i-1/2 , j+1 /2, k-1/2)
4 - (i-1/2 , j-1/ 2, k+1/2)
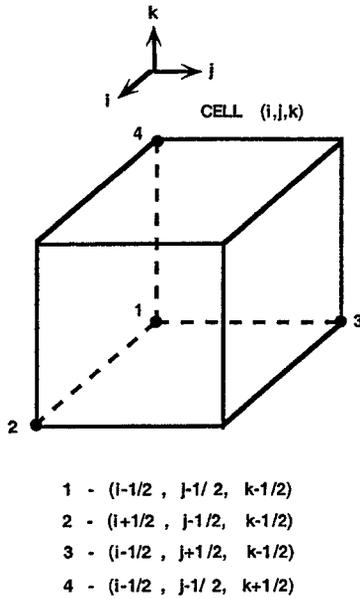
FIG. 1. Global indices for four vertices associated with cell (i, j, k).

The vector and matrix notation used from this point forward in this paper is as follows. Each vector is denoted by an uppercase symbol and the components of that vector are denoted by the corresponding lowercase symbol. An arrow is placed over the uppercase symbol if the vector is physical, while a chevron is placed above the uppercase symbol if the vector is algebraic. Each matrix is denoted by a boldface uppercase symbol and the elements of that matrix are denoted by the corresponding lowercase symbol.

L - (i-1/ 2, j, k)

B - (i, j-1/2, k)

D - (i, j, k-1/2)
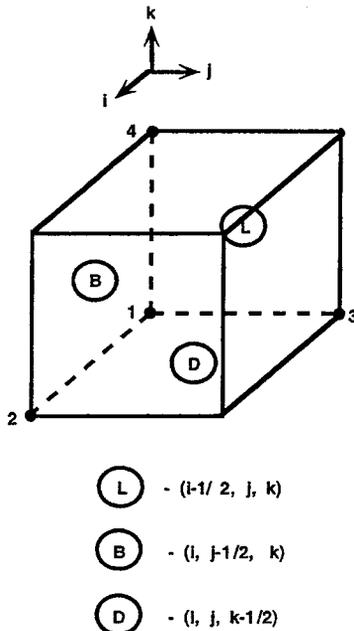
FIG. 2. Local and global indices for three of six face centers associated with cell (i, j, k).
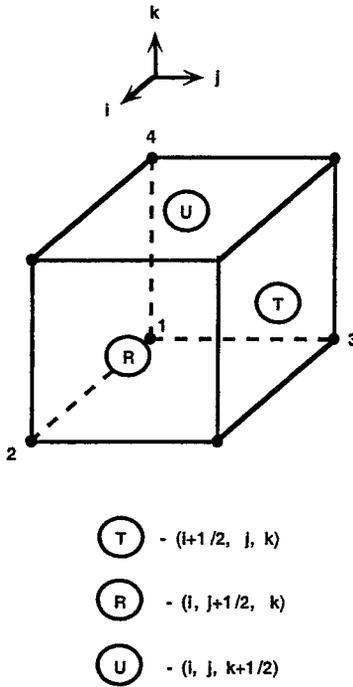
FIG. 3.   Local and global indices for three of six face centers associated with cell (i, j, k).

The intensities (scalars) are defined to exist at both cell center: $\phi_{i,j,k}^C$, and on the face centers: $\phi_{i,j,k}^L$, $\phi_{i,j,k}^R$, $\phi_{i,j,k}^B$, $\phi_{i,j,k}^T$, $\phi_{i,j,k}^D$, $\phi_{i,j,k}^U$. As previously noted, the use of local indices implies that a quantity is uniquely associated with a single cell. For instance, unless it is otherwise stated, one should assume that $\phi_{i,j,k}^R \neq \phi_{i+1,j,k}^L$.

Vectors are defined in terms of face-area components located at the face centers: $f_{i,j,k}^L$, $f_{i,j,k}^R$, $f_{i,j,k}^B$, $f_{i,j,k}^T$, $f_{i,j,k}^D$, $f_{i,j,k}^U$, where $f_{i,j,k}^L$ denotes the dot product of $\vec{F}$ with the outward-directed area vector located at the center of the left face of cell i, j, k. The other face-area components are defined analogously. The area vector is defined as the integral of the
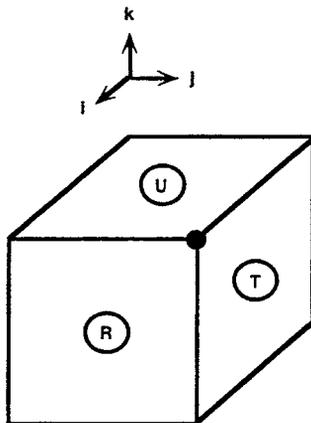
FIG. 4.   Vertex shared by the Right, Top, and Up faces having local index RTU.

outward-directed unit normal vector over the face, i.e.,

$$\vec{A} = \oint \vec{n} \, dA, \tag{7}$$

where $\vec{n}$ is a unit vector that is normal to the faces at each point on the face. The average outward-directed unit normal vector for the face is defined as follows:

$$\langle \vec{n} \rangle = \frac{\vec{A}}{\|\vec{A}\|}, \tag{8}$$

where $\|\vec{A}\|$ denotes the magnitude (standard Euclidian norm) of $\vec{A}$. Equation (8) can be used to convert face-area flux components to face-normal components if desired, e.g.,

$$\vec{F} \cdot \langle \vec{n} \rangle = \vec{F} \cdot \frac{\vec{A}}{\|\vec{A}\|},$$

$$= \frac{f}{\|\vec{A}\|}. \tag{9}$$

Note that $\|\vec{A}\|$ is equal to the face area only when the face is flat. Interestingly, the true face areas never arise in our discretization scheme. Since it takes three components to define a full vector, the full vectors are considered to be located at the cell vertices: $\vec{F}_{i,j,k}^{LBD}, \vec{F}_{i,j,k}^{RBD}, \vec{F}_{i,j,k}^{LTD}, \vec{F}_{i,j,k}^{RTD}, \vec{F}_{i,j,k}^{LBU}, \vec{F}_{i,j,k}^{RBU}, \vec{F}_{i,j,k}^{LTU}, \vec{F}_{i,j,k}^{RTU}$. Each vertex vector is constructed using the face-area components and area vectors associated with the three faces that share that vertex. For instance,

$$\vec{F}_{i,j,k}^{LBD} = \frac{f^L(\vec{A}^B \times \vec{A}^D)}{\vec{A}^L \cdot (\vec{A}^B \times \vec{A}^D)} + \frac{f^B(\vec{A}^D \times \vec{A}^L)}{\vec{A}^L \cdot (\vec{A}^D \times \vec{A}^L)} + \frac{f^D(\vec{A}^L \times \vec{A}^B)}{\vec{A}^D \cdot (\vec{A}^L \times \vec{A}^B)}. \tag{10}$$

It is convenient for our purposes to define an algebraic vector, $\hat{F}$, consisting of the three face-area components associated with the physical vector, $\vec{F}$, e.g.,

$$\hat{F}_{LBD} = \left(f_{i,j,k}^L, f_{i,j,k}^B, f_{i,j,k}^D \right)^t, \tag{11}$$

where a superscript "t" denotes "transpose." The three face-area components associated with the Right-Top-Up vertex are illustrated in Fig. 5. The other vertex vectors are defined in analogy with Eqs. (10) and (11).

The next step in our support-operators method is to discretize Eq. (4) over a single arbitrary cell in a particular manner. Specifically, we explicitly discretize all but the flux operator, which is expressed in an implicit form consistent with our choice of discrete vector unknowns. We assume indices of i, j, k for the arbitrary cell, but suppress these indices whenever possible in the discrete approximation to Eq. (4) that follows. We first discretize the surface integral

$$\oint_{\partial V} \phi \vec{H} \cdot \vec{n} \, dA \approx \phi^L h^L + \phi^R h^R + \phi^B h^B + \phi^T h^T + \phi^D h^D + \phi^U h^U. \tag{12}$$

$$\hat{F} = (\ \vec{F} \bullet \vec{A}_R,\quad \vec{F} \bullet \vec{A}_T,\quad \vec{F} \bullet \vec{A}_U\ )^t$$
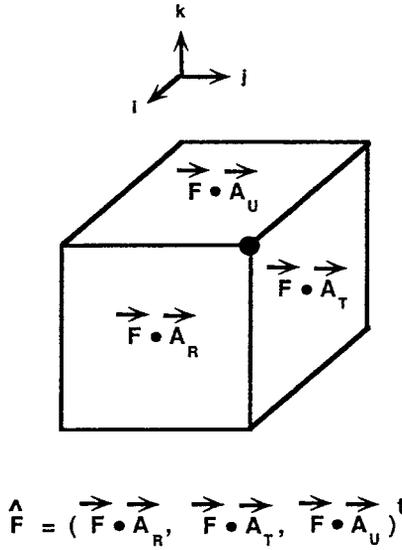
FIG. 5. Three face-center face-area components defining the flux vector at vertex RTU.

Next, we approximate the flux volumetric integral

$$\int_V -D^{-1}\vec{H} \cdot D\vec{\nabla}\phi \, dV$$

$$\approx D^{-1}(\vec{H}^{LBD} \cdot \vec{F}^{LBD})V^{LBD} + D^{-1}(\vec{H}^{RBD} \cdot \vec{F}^{RBD})V^{RBD} + D^{-1}(\vec{H}^{LTD} \cdot \vec{F}^{LTD})V^{LTD}$$

$$+ D^{-1}(\vec{H}^{RTD} \cdot \vec{F}^{RTD})V^{RTD} + D^{-1}(\vec{H}^{LBU} \cdot \vec{F}^{LBU})V^{LBU} + D^{-1}(\vec{H}^{RBU} \cdot \vec{F}^{RBU})V^{RBD}$$

$$+ D^{-1}(\vec{H}^{LTU} \cdot \vec{F}^{LTU})V^{LTU} + D^{-1}(\vec{H}^{RTU} \cdot \vec{F}^{RTU})V^{RTU}, \tag{13}$$

where $\vec{F}^{LBD}$ denotes $-D\vec{\nabla}\phi$ at the Left-Bottom-Down vertex, and V^{LBD} denotes the volumetric weight associated with the Left-Bottom-Down vertex. The remaining flux vectors and vertex volumetric weights are analogously indexed. The choice of weights is one of the many free parameters in the support-operators method. For reasons explained in Section 3, we define our weights in terms of triple products. Specifically, each vertex weight is initially given by one-eighth the triple product associated with the vertex. For instance, using the local vertex indexing shown in Fig. 2, the volumetric weight for the Left-Bottom-Down vertex is given by

$$V^{LBD} = \frac{1}{8}\vec{R}_{1,2} \times \vec{R}_{1,3} \cdot \vec{R}_{1,4}, \tag{14}$$

where $\vec{R}_{i,j}$ denotes the vector from vertex i to vertex j. These triple product weights do not sum to the total volume of the hexahedron unless the hexahedron is a parallelepiped. Thus, we normalize the triple product weights so that they sum to the correct total volume. In particular, each triple-product weight is multiplied by the ratio V_t/V_s, where V_t is the total volume of the hexahedron and V_s is the sum of the triple product weights.

One can evaluate the dot products in Eq. (13) using Eq. (10), but we find it better for our purposes to evaluate them with the algebraic face-area flux vectors defined by Eq. (11). This

is achieved by first transforming the face-area vectors to Cartesian vectors and then taking the dot product. Rather than explicitly defining the matrix that transforms face-area vectors to Cartesian vectors, we explicitly define its inverse. The desired transformation matrix can then be obtained by either algebraic or numerical inversion. For instance, let us consider the Left-Bottom-Down vertex vectors. We denote the matrix that transforms face-area vectors to Cartesian vectors as A^{LBD}. Its inverse is the matrix that transforms Cartesian vectors to face-area vectors,

$$\hat{H}^{LBD} = [\mathbf{A}^{LBD}]^{-1}\vec{H}^{LBD}, \tag{15}$$

where $\hat{H}$ denotes a Left-Bottom-Down face-area flux vector,

$$\hat{H} = (h^L, h^B, h^D)^t, \tag{16}$$

and $\vec{H}$ denotes a Left-Bottom-Down Cartesian flux vector,

$$\vec{H} = (h^x, h^y, h^z)^t, \tag{17}$$

and

$$[\mathbf{A}^{LBD}]^{-1} = \begin{bmatrix} a_x^L & a_y^L & a_z^L \\ a_x^B & a_y^B & a_z^B \\ a_x^D & a_y^D & a_z^D \end{bmatrix}, \tag{18}$$

where a_x^L denotes the x-component of the area vector associated with the left face. The remaining components of the matrix are defined analogously. Transforming the face-area vector for the Left-Bottom-Down vertex, we obtain

$$\vec{H}^{LBD} \cdot \vec{F}^{LBD} = \mathbf{A}^{LBD}\hat{H}^{LBD} \cdot \mathbf{A}^{LBD}\hat{F}^{LBD},$$
$$= \hat{H}^{LBD} \cdot \mathbf{S}^{LBD}\hat{F}^{LBD}, \tag{19}$$

where

$$\mathbf{S}^{LBD} = [\mathbf{A}^{LBD}]^t\mathbf{A}^{LBD}. \tag{20}$$

Following Eq. (20), we now rewrite Eq. (13) in terms of face-area vectors as

$$\int_V -D^{-1}\vec{H} \cdot D\vec{\nabla}\phi\, dV$$
$$\approx D^{-1}(\hat{H}^{LBD} \cdot \mathbf{S}^{LBD}\hat{F}^{LBD})V^{LBD} + D^{-1}(\hat{H}^{RBD} \cdot \mathbf{S}^{RBD}\hat{F}^{RBD})V^{RBD}$$
$$+ D^{-1}(\hat{H}^{LTD} \cdot \mathbf{S}^{LTD}\hat{F}^{LTD})V^{LTD} + D^{-1}(\hat{H}^{RTD} \cdot \mathbf{S}^{RTD}\hat{F}^{RTD})V^{RTD}$$
$$+ D^{-1}(\hat{H}^{LBU} \cdot \mathbf{S}^{LBU}\hat{F}^{LBU})V^{LBU} + D^{-1}(\hat{H}^{RBU} \cdot \mathbf{S}^{RBU}\hat{F}^{RBU})V^{RBU}$$
$$+ D^{-1}(\hat{H}^{LTU} \cdot \mathbf{S}^{LTU}\hat{F}^{LTU})V^{LTU} + D^{-1}(\hat{H}^{RTU} \cdot \mathbf{S}^{LTU}\hat{F}^{RTU})V^{RTU}. \tag{21}$$

Although we assume a single diffusion coefficient in each cell in this paper, we note that our scheme can accommodate a different diffusion coefficient for each vertex. In particular, Eq. (21) becomes

$$
\begin{aligned}
\int_V & -D^{-1}\vec{H} \cdot D\vec{\nabla}\phi \, dV \\
&\approx D^{LBD^{-1}}(\hat{H}^{LBD} \cdot \mathbf{S}^{LBD}\hat{F}^{LBD})V^{LBD} + D^{RBD^{-1}}(\hat{H}^{RBD} \cdot \mathbf{S}^{RBD}\hat{F}^{RBD})V^{RBD} \\
&\quad + D^{LTD^{-1}}(\hat{H}^{LTD} \cdot \mathbf{S}^{LTD}\hat{F}^{LTD})V^{LTD} + D^{RTD^{-1}}(\hat{H}^{RTD} \cdot \mathbf{S}^{RTD}\hat{F}^{RTD})V^{RTD} \\
&\quad + D^{LBU^{-1}}(\hat{H}^{LBU} \cdot \mathbf{S}^{LBU}\hat{F}^{LBU})V^{LBU} + D^{RBU^{-1}}(\hat{H}^{RBU} \cdot \mathbf{S}^{RBU}\hat{F}^{RBU})V^{RBU} \\
&\quad + D^{LTU^{-1}}(\hat{H}^{LTU} \cdot \mathbf{S}^{LTU}\hat{F}^{LTU})V^{LTU} + D^{RTU^{-1}}(\hat{H}^{RTU} \cdot \mathbf{S}^{LTU}\hat{F}^{RTU})V^{RTU}. \quad (22)
\end{aligned}
$$

Although we assume a scalar diffusion coefficient in this paper, we note that our scheme can accommodate a tensor diffusion coefficient. Specifically, with a tensor diffusion coefficient at each vertex, Eq. (21) becomes

$$
\begin{aligned}
\int_V & -D^{-1}\vec{H} \cdot D\vec{\nabla}\phi \, dV \\
&\approx (\hat{H}^{LBD} \cdot \mathbf{G}^{LBD}\hat{F}^{LBD})V^{LBD} + (\hat{H}^{RBD} \cdot \mathbf{G}^{RBD}\hat{F}^{RBD})V^{RBD} + (\hat{H}^{LTD} \cdot \mathbf{G}^{LTD}\hat{F}^{LTD})V^{LTD} \\
&\quad + (\hat{H}^{RTD} \cdot \mathbf{G}^{RTD}\hat{F}^{RTD})V^{RTD} + (\hat{H}^{LBU} \cdot \mathbf{G}^{LBU}\hat{F}^{LBU})V^{LBU} + (\hat{H}^{RBU} \cdot \mathbf{G}^{RBU}\hat{F}^{RBU})V^{RBU} \\
&\quad + (\hat{H}^{LTU} \cdot \mathbf{G}^{LTU}\hat{F}^{LTU})V^{LTU} + (\hat{H}^{RTU} \cdot \mathbf{G}^{LTU}\hat{F}^{RTU})V^{RTU}, \quad (23)
\end{aligned}
$$

where

$$
\mathbf{G}^{LBD} = [\mathbf{A}^{LBD}]^t[\mathbf{D}^{LBD}]^{-1}\mathbf{A}^{LBD}, \quad (24)
$$

and $\mathbf{D}^{LBD}$ is the Left-Bottom-Down diffusion tensor in the Cartesian basis. The remaining **G**-matrices are defined analogously. The diffusion tensor must be symmetric positive-definite to ensure that its inverse exists and that the coefficient matrix for our diffusion scheme is symmetric positive-definite.

Finally, we approximate the divergence volumetric integral

$$
\int_V \phi\vec{\nabla} \cdot \vec{H} \, dV \approx \phi^C[h^L + h^R + h^B + h^T + h^D + h^U]. \quad (25)
$$

Equations (12), (21), and (25) are certainly not unique, but they are fairly straightforward. For instance, Eq. (12) represents a face-centered second-order approximation to a surface integral. Equation (21) represents a vertex-based volumetric integral consisting of a dot-product contribution from each pair of vertex vectors. Equation (25) is a particularly simple second-order approximation which gives all of the weight to the cell-center value of ϕ while using a surface-integral formulation for $\vec{\nabla} \cdot \vec{H}$ that is analogous to the surface-integral used in Eq. (12).

Substituting from Eqs. (12), (21), and (25) into Eq. (4), we obtain the discrete version of Eq. (4):

$$\phi^L h^L + \phi^R h^R + \phi^B h^B + \phi^T h^T + \phi^D h^D + \phi^U h^U + D^{-1}(\hat{H}^{LBD} \cdot \mathbf{S}^{LBD}\hat{F}^{LBD})V^{LBD}$$

$$+ D^{-1}(\hat{H}^{RBD} \cdot \mathbf{S}^{RBD}\hat{F}^{RBD})V^{RBD} + D^{-1}(\hat{H}^{LTD} \cdot \mathbf{S}^{LTD}\hat{F}^{LTD})V^{LTD}$$

$$+ D^{-1}(\vec{H}^{RTD} \cdot \mathbf{S}^{RTD}\hat{F}^{RTD})V^{RTD} + D^{-1}(\hat{H}^{LBU} \cdot \mathbf{S}^{LBU}\hat{F}^{LBU})V^{LBU}$$

$$+ D^{-1}(\hat{H}^{RBU} \cdot \mathbf{S}^{RBU}\hat{F}^{RBU})V^{RBU} + D^{-1}(\hat{H}^{LTU} \cdot \mathbf{S}^{LTU}\hat{F}^{LTU})V^{LTU}$$

$$+ D^{-1}(\hat{H}^{RTU} \cdot \mathbf{S}^{LTU}\hat{F}^{RTU})V^{RTU} = \phi^C[h^L + h^R + h^B + h^T + h^D + h^U]. \tag{26}$$

Note that Eq. (26) defines the discrete inner products, discussed in [10], that are associated with the adjoint relationship between the divergence and gradient operators. We can now use this relationship to solve for the flux operator components by requiring that the resulting discretized identity hold for *all* discrete $\hat{H}$ values. In particular, the equation for the face-area component of $\vec{F}$ on any given cell face is obtained from Eq. (26) simply by setting the same face-area component of $\vec{H}$ on that face to unity and setting the remaining face-area components of $\vec{H}$ on all the other faces to zero. For instance, we obtain the equation for f^L from Eq. (26) by setting h^L to unity and all the other face-area components of $\vec{H}$, i.e., h^R, h^B, h^T, h^D, h^U, to zero as

$$\phi^L + D^{-1}\left(s_{L,L}^{LBD}f^L + s_{L,B}^{LBD}f^B + s_{L,D}^{LBD}f^D\right)V^{LBD} + D^{-1}\left(s_{L,L}^{LTD}f^L + s_{L,T}^{LTD}f^T + s_{L,D}^{LTD}f^D\right)V^{LTD}$$

$$+ D^{-1}\left(s_{L,L}^{LBU}f^L + s_{L,B}^{LBU}f^B + s_{L,U}^{LBU}f^U\right)V^{LBU} + D^{-1}\left(s_{L,L}^{LTU}f^L + s_{L,T}^{LTU}f^T\right.$$

$$\left. + s_{L,U}^{LTU}f^U\right)V^{LTU} = \phi^C, \tag{27}$$

where $s_{L,L}^{LBD}$ denotes the (L, L) element of the matrix $\mathbf{S}^{LBD}$ defind by Eq. (20), and the remaining $\mathbf{S}$-matrix elements are defined analogously.

The equations for the face-area flux components, i.e., Eq. (27) and its analogues for the Right, Bottom/Top, and Down/Up faces, can be expressed in matrix from as

$$\mathbf{W}^{-1}\hat{\mathcal{F}} = \Delta\hat{\Phi}, \tag{28}$$

where

$$\hat{\mathcal{F}} = (f^L, f^R, f^B, f^T, f^D, f^U)^t, \tag{29}$$

and

$$\Delta\hat{\Phi} = (\phi^C - \phi^L, \phi^C - \phi^R, \phi^C - \phi^B, \phi^C - \phi^T, \phi^C - \phi^D, \phi^C - \phi^U)^t. \tag{30}$$

To obtain a matrix that gives the face-center components of the flux operator in terms of the face-center and cell-center intensities, one need simply invert the 6×6 matrix in Eq. (28):

$$\hat{\mathcal{F}} = \mathbf{W}\Delta\hat{\Phi}. \tag{31}$$

Since it is not practical to perform this inversion algebraically, we perform it numerically. Thus, we cannot give an explicit expression for the matrix $\mathbf{W}$. Nonetheless, it can be shown that it is an SPD matrix (see the Appendix). In addition, if we assume an orthogonal mesh, $\mathbf{W}$ becomes diagonal and can be trivially inverted. For instance, under this assumption,

Eq. (27) becomes

$$\phi^L + D^{-1}(\Delta y \Delta z)^{-2} f^L \frac{\Delta x \Delta y \Delta z}{2} = \phi_C, \tag{32}$$

where we have also assumed that the indices i, j, k, correspond to the spatial coordinates x, y, z, respectively. Solving Eq. (32) for f^L, we obtain

$$f^L = -\frac{2D}{\Delta x}(\phi^L - \phi^C)\Delta y \Delta z, \tag{33}$$

which is exact for a ϕ that is linearly dependent upon x.

Having derived Eq. (31), we can construct the discrete equation for the cell-center intensity in every cell. Each such equation represents a discretization of Eq. (3), i.e., a balance equation for the cell. Furthermore, each balance equation uses a discretization for the divergence of the flux that is identical to that used in Eq. (26). In some sense, this is the point at which we obtain a diffusion operator by combining our discrete divergence and flux operators. Specifically, the equation for ϕ^C is

$$f^L + f^R + f^B + f^T + f^D + f^U = Q^C V, \tag{34}$$

where V denotes the total volume of the cell, the face-area flux components are expressed in terms of the intensities via Eq. (31), and Q^C denotes the source or driving function evaluated at cell-center. Equation (34) contains all of the intensities in cell (i, j, k). Thus, it has a 7-point stencil.

Now that we have defined the equations for the cell-center intensities, we must next define equations for the face-center intensities. Our local indexing scheme admits two intensities and two face-area flux components at each face on the mesh interior. In particular, there is one intensity and one flux component from each of the cells that share a face. For instance, the cell face with global index $(i + \frac{1}{2}, j, k)$ is associated with the two intensities, $\phi^R_{i,j,k}$ and $\phi^L_{i+1,j,k}$, and the two face-area flux components, $f^R_{i,j,k}$ and $f^L_{i+1,j,k}$. We previously obtained the flux components in terms of the intensities by forcing Eq. (26), a discrete version of Eq. (4), to be satisfied on each individual cell for all discrete scalars and vectors. We now obtain equations for the interior-mesh face-center intensities by requiring that this identity be satisfied over the entire mesh for all discrete scalars and vectors.

When Eq. (26) is summed over the entire mesh, the two volumetric integrals are naturally approximated in terms of a sum of contributions from each individual cell. However, a valid approximation for the surface integral in Eq. (26) will occur if and only if contributions to the surface integral from each individual cell cancel at all interior faces, thereby resulting in an approximate integral over the outer surface of the mesh. By inspection of Eq. (26), it can be seen that this will be achieved by requiring both continuity of the intensity and continuity of the face-area flux component at each interior cell face. In particular, we require that

$$\phi^R_{i,j,k} = \phi^L_{i+1,j,k} \equiv \phi_{i+\frac{1}{2},j,k}, \tag{35}$$

$$\phi^T_{i,j,k} = \phi^B_{i,j+1,k} \equiv \phi_{i,j+\frac{1}{2},k}, \tag{36}$$

$$\phi^U_{i,j,k} = \phi^D_{i,j,k+1} \equiv \phi_{i,j,k+\frac{1}{2}}, \tag{37}$$

$$f_{i,j,k}^R + f_{i+1,j,k}^L = 0, \tag{38}$$

$$f_{i,j,k}^T + f_{i,j+1,k}^B = 0, \tag{39}$$

$$f_{i,j,k}^U + f_{i,j,k+1}^D = 0, \tag{40}$$

where the indices in Eqs. (35) through (40) take on all values associated with interior cell faces, and the flux components in Eqs. (38) through (40) are expressed in terms of intensities via Eq. (31). One would expect that the continuity of the face-area flux components expressed by Eqs. (38) through (40) would require that the *difference* of the components be zero rather than the *sum* of the components. However, one must remember that each of the components is defined with respect to an area vector that is equal in magnitude but opposite in direction to that of the other component.

Equations (35) through (37) establish that there is only one intensity unknown associated with each interior-mesh cell face. Thus, as shown in Eqs. (35) through (37), each such intensity can be uniquely referred to using a global mesh index. The equations for these intensities are given by Eqs. (38) through (40). For instance, Eq. (38) is the equation for $\phi_{i+\frac{1}{2},j,k}$. In general, Eq. (38) contains only and all of the intensities in cells (i, j, k) and $(i + 1, j, k)$. Thus, it has a 13-point stencil. The only intensity shared by these two cells is $\phi_{i+\frac{1}{2},j,k}$. Thus, in a certain sense it can be said that $\phi_{i+\frac{1}{2},j,k}$ is "chosen" to obtain continuity of the face-area flux components on cell-face $(i + \frac{1}{2}, j, k)$. The properties of Eqs. (39) and (40) are completely analogous to those of Eq. (38).

If the mesh is orthogonal, Eqs. (38) through (40) simplify to such an extent that they relate each interior-mesh face-center intensity to the two cell-center intensities adjacent to it. This enables the face-center intensities to be explicitly eliminated, resulting in the standard 7-point cell-centered diffusion discretization that is both SPD and monotone (having strictly positive diagonal elements and strictly nonpositive off-diagonal elements.) This is completely analogous to the 2-D case discussed in detail in [2]. However, if the mesh is nonorthogonal, the face-center intensities cannot be eliminated, and Eqs. (38) through (40) must be included in the diffusion matrix. In this case, these equations must be reversed in sign to obtain a symmetric diffusion matrix,

$$-f_{i,j,k}^R - f_{i+1,j,k}^L = 0, \tag{41}$$

$$-f_{i,j,k}^T - f_{i,j+1,k}^B = 0, \tag{42}$$

$$-f_{i,j,k}^U - f_{i,j,k+1}^D = 0. \tag{43}$$

Having defined the equations for the cell-center and interior-mesh face-center intensities, we need only define the equations for the face-center intensities on the outer mesh boundary to complete the specification of our diffusion discretization scheme. Cell faces on the outer boundary are associated with only one cell. Thus, there is only one face-center intensity and one face-area flux component associated with each such face. The equation for each boundary intensity is very similar to that for each interior-mesh face-center intensity in that it expresses a continuity of the face-normal flux component. The only difference in the boundary equations is that the analytic boundary condition for the diffusion equation is used to define a "ghost-cell" face-normal flux component that must be equated to the standard face-normal flux component defined by Eq. (31). A ghost cell is a nonexistent mesh cell that represents a continuation of the mesh across the outer mesh boundary. For instance, assuming that the left face of cell $1, j, k$ is on the outer boundary of the mesh and

its remaining faces are on the interior of the mesh, the ghost cell "adjacent" to cell $1, j, k$ carries the indices $0, j, k$.

The analytic diffusion boundary condition of interest to us is the so-called "extrapolated" boundary condition. This condition is of the mixed or Robin type and can be expressed as

$$\phi + d^e \vec{\nabla}\phi \cdot \vec{n} = \phi^e, \tag{44}$$

where d^e is called the extrapolation distance, ϕ^e is called the extrapolated intensity (a specified function), and $\vec{n}$ denotes an outward-directed unit normal vector. Equation (44) is satisfied at each point on the outer surface of the problem domain. Of course, the values of the parameters d^e and ϕ^e may change as a function of position. One obtains a vacuum boundary condition when $\phi^e = 0$, a source condition when ϕ^e is nonzero, and a reflective (Neumann) condition when $\phi^e = \phi$. The extrapolated boundary condition is said to be a Marshak condition whenever $d^e = 2D$.

We begin the derivation of the ghost-cell face-area flux component by substituting from Eq. (2) into Eq. (44)

$$\phi - \frac{d^e}{D}\vec{F}^g \cdot \vec{n} = \phi^e, \tag{45}$$

where $\vec{F}$ is the flux vector associated with a ghost cell. Next we recognize that the outward-directed unit normal vector for a ghost-cell must be identical to an inward-directed unit normal vector on the outer surface of the problem domain. Thus,

$$\vec{n}^g = -\vec{n}, \tag{46}$$

where $\vec{n}^g$ denotes a ghost-cell outward-directed unit normal vector. Substituting from Eq. (46) into Eq. (45), we obtain

$$\phi + \frac{d^e}{D}\vec{F}^g \cdot \vec{n}^g = \phi^e, \tag{47}$$

Next, we solve Eq. (47) for the outward-directed flux component associated with a ghost cell:

$$\vec{F}^g \cdot \vec{n}^g = \frac{D}{d^e}(\phi^e - \phi). \tag{48}$$

Now let us assume that the left face of cell $1, j, k$ is on the outer boundary of the mesh with its remaining faces on the mesh interior. The ghost cell whose right face is identical to the left face of cell $1, j, k$ carries the indices $0, j, k$. The intensity on the left face of cell $(1, j, k)$ is $\phi_{\frac{1}{2},j,k}$, and the face-area flux component on that face is $f^L_{1,j,k}$. Evaluating Eq. (48) at the center of face $(\frac{1}{2}, j, k)$ and multiplying the resulting expression by the magnitude of the outward-directed area-vector on that face associated with cell $1, j, k$, we obtain the desired expression for the ghost-cell face-area flux component,

$$f^R_{0,j,k} = -\frac{D_{1,j,k}}{d^e_{0,j,k}}\left(\phi_{\frac{1}{2},j,k} - \phi^e_{0,j,k}\right)\|\vec{A}^L_{1,j,k}\|, \tag{49}$$

where the extrapolated intensity and the extrapolation distance are assumed to carry the ghost-cell index.

We next obtain the equation for $\phi_{\frac{1}{2},j,k}$ by requiring that the Right and Left face-area flux components for cells $(0, j, k)$ and $(1, j, k)$, respectively, sum to zero:

$$-f_{0,j,k}^{R} - f_{1,j,k}^{L} = 0. \tag{50}$$

Note that Eq. (50) is identical to Eq. (41) with the latter equation evaluated at $i = 0$. Thus, Eqs. (41) through (43) provide *all* face-center intensity equations with the caveat that when an intensity is on the outer mesh boundary, the associated ghost-cell flux component must be defined via the boundary condition rather than Eq. (31). Note that Eq. (50) couples all of the intensities within a cell and therefore has a 7-point stencil. This completes the specification of our diffusion discretization scheme.

To summarize,

- The face-area flux components for each cell are expressed in terms of the intensities within that cell via Eq. (31).
- The discrete equation for each cell-centered intensity is given in Eq. (34).
- The equations for the interior-mesh face-centered intensities are given in Eqs. (41) through (43).
- The equation for a face-center intensity on the outer mesh boundary is given by Eqs. (49) and (50) when the boundary face is the Left face of a cell. Analogous equations for the other five cases are easily derived using Eqs. (41) through (43) and Eq. (49).

We have already shown that our diffusion matrix is sparse. It is also symmetric positive-definite. We demonstrate this latter property in the Appendix. If the mesh is orthogonal, the **W**-matrices, defined by Eq. (31), become diagonal, and the face-center intensities can be locally eliminated from the cell-center/face-center system. This results in a pure cell-center diffusion discretization that is identical to the standard 7-point cell-center diffusion discretization scheme [8]. In Section 4 we describe a preconditioner that exploits the fact that the face-center intensities can be locally eliminated when the **W**-matrices are diagonal.

3. CHOICES IN THE SUPPORT-OPERATORS METHOD

In this section we discuss some of the choices that we have made in formulating our support-operators method. As previously noted, the general principle upon which the support-operators method is based can be applied with a variety of locations and representations for the intensity and flux unknowns. There is a great deal of freedom in the choice of numerical parameters used in evaluating the volumetric and surface integrals. However, the accuracy of a scheme is highly dependent upon the choices that are made.

For instance, rather than define the fluxes in terms of face-normal components on each face, one can simply place an independent set of Cartesian flux components at each vertex. This yields a total of 24 flux unknowns for a hexahedron rather than the six unknowns in our scheme. However, schemes constructed in this way generally do not converge on nonsmooth meshes. The use of face-normal components appears to be necessary to achieve convergence on such meshes.

On a general hexahedron, there is even some choice in defining the face-normal components. For instance, the surface normal varies across a hexahedral face. Thus, one must
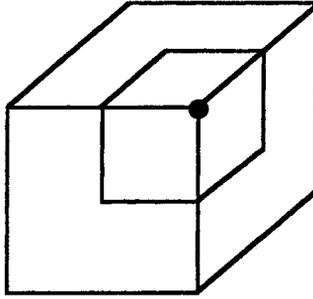
FIG. 6. Sub-hexahedron associated with vertex.

choose a particular representation for the "average face normal." We have chosen to define this quantity in terms of the area vector, but other choices are clearly possible.

The vertex volumetric weights used in the flux volumetric integral can also be chosen in a variety of ways. We have investigated alternatives to the normalized triple-product weights. In particular, we considered unnormalized triple-product weights, sub-hexahedron weights, and simple one-eighth weights. Sub-hexahedron weights are defined by the volume of an associated sub-hexahedron. The sub-hexahedra are obtained by using four straight lines to connect each face center with the four edge centers adjacent to it, and by using six straight lines to connect the cell center with the six face centers. A sub-hexahedron is illustrated in Fig. 6. Although it may not be obvious, each other face of each sub-hexahedron coincides with a face of the hexahedron. Thus, the volumes of the sub-hexahedra always sum to the total hexahedron volume. This would seem to be the most natural choice for the vertex volumetric weights. The one-eighth weights are identical for a given hexahedron and are obtained by dividing the total volume of the hexahedron by one-eighth.

Computational testing indicates that the sub-hexagon and one-eighth weights are decidedly inferior to the triple-product and normalized triple-product weights. In particular, the triple-product and normalized triple-product weights both yield a second-order-accurate diffusion discretization, whereas the sub-hexagon and one-eighth weights yield a first-order accurate diffusion discretization. Although they both give second-order accuracy, the normalized triple-product weights seem to be slightly more accurate than the triple product weights. This is why we use the normalized triple-product weights.

4. SOLUTION OF THE EQUATIONS

We use a preconditioned conjugate-gradient method [9] to solve our discretized diffusion equations. The preconditioner is completely analogous to that used for the 2-D local support-operators scheme [2]. It is obtained simply by setting the off-diagonal elements of the **S**-matrices, defined by Eq. (20), to zero. This causes the **W**-matrices, defined by Eq. (31), to be diagonal and effects a huge simplification in the algebraic structure of the intensity equations. In particular, this makes it possible to locally eliminate the face-center intensities from the cell-center/face-center system, resulting in a pure cell-center diffusion discretization that is both SPD and monotone. For instance, if we set the off-diagonal elements of the **S**-matrices to zero, Eq. (27) and its analogue for f^R yield

$$f^L_{i+1,j,k} = -\frac{2D_{i+1,j,k}}{\Delta^L_{i+1,j,k}}\left(\phi_{i+\frac{1}{2},j,k} - \phi_{i+1,j,k}\right), \tag{51}$$

and

$$f_{i,j,k}^{R} = -\frac{2D_{i,j,k}}{\Delta_{i,j,k}^{R}}\left(\phi_{i+\frac{1}{2},j,k} - \phi_{i,j,k}\right), \tag{52}$$

respectively, where

$$\Delta_{i+1,j,k}^{L} = 2\left[s_{L,L}^{LBD}V^{LBD} + s_{L,L}^{LTD}V^{LTD} + s_{L,L}^{LBU}V^{LBU} + s_{L,L}^{LTU}V^{LTU}\right]_{i+1,j,k}, \tag{53}$$

and

$$\Delta_{i,j,k}^{R} = 2\left[s_{R,R}^{RBD}V^{RBD} + s_{R,R}^{RTD}V^{RTD} + s_{R,R}^{RBU}V^{RBU} + s_{R,R}^{RTU}V^{RTU}\right]_{i,j,k}. \tag{54}$$

Substituting from Eqs. (51) and (52), into Eq. (41), we get the equation for $\phi_{i+\frac{1}{2},j,k}$:

$$\frac{2D_{i,j,k}\left(\phi_{i+\frac{1}{2},j,k} - \phi_{i,j,k}\right)}{\Delta_{i,j,k}^{R}} + \frac{2D_{i+1,j,k}\left(\phi_{i+\frac{1}{2},j,k} - \phi_{i+1,j,k}\right)}{\Delta_{i+1,j,k}^{L}} = 0. \tag{55}$$

Solving Eq. (55) for $\phi_{i+\frac{1}{2},j,k}$, we get

$$\phi_{i+\frac{1}{2},j,k} = \left(\phi_{i,j,k}\frac{D_{i,j,k}}{\Delta_{i,j,k}^{R}} + \phi_{i+1,j,k}\frac{D_{i+1,j,k}}{\Delta_{i+1,j,k}^{L}}\right)\bigg/\left(\frac{D_{i,j,k}}{\Delta_{i,j,k}^{R}} + \frac{D_{i+1,j,k}}{\Delta_{i+1,j,k}^{L}}\right). \tag{56}$$

Thus, we see from Eq. (56) that neglecting the off-diagonal elements of the **S**-matrices makes each interior-mesh face-center intensity a weighted-average of the two cell-center intensities adjacent to it. Substituting from Eq. (56) into Eqs. (51) and (52) we find that the face-area fluxes on the right and left faces of cells (i, j, k) and $(i + 1, j, k)$, respectively, can be expressed in terms of a difference between the cell-center intensities in those two cells,

$$f_{i,j,k}^{R} = -f_{i+1,j,k}^{L} = -\frac{D_{i+\frac{1}{2},j,k}}{\Delta_{i+\frac{1}{2},j,k}}(\phi_{i+1,j,k} - \phi_{i,j,k}), \tag{57}$$

where

$$D_{i+\frac{1}{2},j,k} = \left[\left(\frac{\Delta_{i,j,k}^{R}}{D_{i,j,k}} + \frac{\Delta_{i+1,j,k}^{L}}{D_{i+1,j,k}}\right)\bigg/\left(\Delta_{i,j,k}^{R} + \Delta_{i+1,j,k}^{L}\right)\right]^{-1}, \tag{58}$$

and

$$\Delta_{i+\frac{1}{2},j,k} = \frac{\Delta_{i,j,k}^{R} + \Delta_{i+1,j,k}^{L}}{2}. \tag{59}$$

Thus, each interior-mesh face-area flux can be expressed in terms of a difference between the two adjacent cell-center intensities. Substituting from Eq. (57) (and its analogues for the other face-area fluxes) into the balance equation, Eq. (34), we obtain a 7-point cell-center

diffusion discretization for each cell on the mesh interior. In particular, the balance equation for cell (i, j, k) (and the equation for $\phi_{i,j,k}$) is

$$-\frac{D_{i+\frac{1}{2},j,k}}{\Delta_{i+\frac{1}{2},j,k}}(\phi_{i+1,j,k} - \phi_{i,j,k}) + \frac{D_{i-\frac{1}{2},j,k}}{\Delta_{i-\frac{1}{2},j,k}}(\phi_{i,j,k} - \phi_{i-1,j,k}) - \frac{D_{i,j+\frac{1}{2},k}}{\Delta_{i,j+\frac{1}{2},k}}(\phi_{i,j+1,k} - \phi_{i,j,k})$$

$$+\frac{D_{i,j-\frac{1}{2},k}}{\Delta_{i,j-\frac{1}{2},k}}(\phi_{i,j,k} - \phi_{i,j-1,k}) - \frac{D_{i,j,k+\frac{1}{2}}}{\Delta_{i,j,k+\frac{1}{2}}}(\phi_{i,j,k+1} - \phi_{i,j,k})$$

$$+\frac{D_{i,j,k-\frac{1}{2}}}{\Delta_{i,j,k-\frac{1}{2}}}(\phi_{i,j,k} - \phi_{i,j,k-1}) = Q_{i,j,k}V_{i,j,k}. \tag{60}$$

To obtain the analogue of Eq. (57) for a cell face on the outer mesh boundary, we again consider a cell $(1, j, k)$, whose left face is on the boundary with its other faces in the mesh interior. Substituting from Eqs. (49) and (51) into Eq. (50), we obtain the equation for $\phi_{\frac{1}{2},j,k}$,

$$\frac{2D_{1,j,k}}{\Delta_{0,j,k}^{R}}\left(\phi_{\frac{1}{2},j,k} - \phi_{0,j,k}^{e}\right) + \frac{2D_{1,j,k}}{\Delta_{1,j,k}^{L}}\left(\phi_{\frac{1}{2},j,k} - \phi_{1,j,k}\right) = 0, \tag{61}$$

where

$$\Delta_{0,j,k}^{R} = \frac{2d_{0,j,k}^{e}}{\|\vec{A}^{L}_{1,j,k}\|}. \tag{62}$$

Solving Eq. (61) for $\phi_{\frac{1}{2},j,k}$ we get

$$\phi_{\frac{1}{2},j,k} = \left(\phi_{0,j,k}^{e}\frac{D_{1,j,k}}{\Delta_{0,j,k}^{R}} + \phi_{1,j,k}\frac{D_{1,j,k}}{\Delta_{1,j,k}^{L}}\right) \Big/ \left(\frac{D_{1,j,k}}{\Delta_{0,j,k}^{R}} + \frac{D_{1,j,k}}{\Delta_{1,j,k}^{L}}\right). \tag{63}$$

Substituting from Eq. (63) into Eqs. (49) and (51), respectively, we obtain the desired expression for the face-area flux component on a boundary face,

$$f_{0,j,k}^{R} = -f_{1,j,k}^{L} = -\frac{D_{1,j,k}}{\Delta_{\frac{1}{2},j,k}}\left(\phi_{\frac{1}{2},j,k} - \phi_{0,j,k}^{e}\right), \tag{64}$$

where $\Delta_{\frac{1}{2},j,k}$ is given by Eq. (59) evaluated with $i = 0$ and Eq. (62). This completes the derivation of the approximate cell-center diffusion scheme used to precondition the full cell-center/face-center scheme.

To summarize:

• The preconditioning system is obtained simply by setting the off-diagonal elements of the **S**-matrices to zero.

• Having diagonal **S**-matrices enables the face-center intensities to be locally eliminated, resulting in a pure 7-point cell-center diffusion discretization on the mesh interior that is given by Eq. (60). Equations (57) and (64) together with their analogs for the Bottom/Top and Down/Up face-area fluxes are used in conjunction with the balance equation to obtain the analogue of Eq. (60) for boundary cells.

• Once the reduced system has been solved for the cell-center intensities, the face-center intensities can be directly calculated. In particular, Eq. (56) and its analogues for

the Bottom/Top and Down/Up faces are used to calculate the face-center intensities on the mesh interior, while Eq. (63) and its analogues for the Bottom/Top and Down/Up faces are used to calculate the face-center intensities on the outer mesh boundary.

Since the **S**-matrices are rigorously diagonal when the mesh is orthogonal, it follows that the preconditioning system is identical to the full cell-center/face-center system whenever the mesh is orthogonal. Thus, our preconditioner can be expected to be very effective if the mesh is not too skewed. Our preconditioning system costs much less to solve than the full system because the coefficient matrix of the reduced cell-center preconditioning system has roughly one-fourth as many rows and one-sixth as many elements as the full cell-center/face-center coefficient matrix. Computational results presented in the next section confirm this expectation.

When the 7-point system is used for preconditioning purposes, an inhomogeneous source term (actually a residual) will generally appear in both the cell-center and face-center intensity equations. We did not include such a source in our derivation of the face-center intensity equations because they do not appear in standard calculations. One must remember to include these sources *before* the face-center intensities are locally eliminated to obtain the 7-point cell-center system. This matter is extensively discussed for the 2-D case in [2].

It can be shown that the cell-center/face-center preconditioning system and the reduced cell-center system are both SPD and monotone. For instance, the demonstration of the SPD property given in the Appendix for the full cell-center/face-center system also applies to the preconditioner. Monotonicity is fairly easy to demonstrate once it is recognized that the "Δ-coefficients" defined by Eqs. (53), (54), (59), and (62) are always positive. This follows from the structure of the **S**-matrices shown in Eq. (20).

5. COMPUTATIONAL RESULTS

In this section we perform four sets of calculations. The first, second, and third sets demonstrate convergence properties of our method on both well-behaved nonsmooth grids and ill-behaved highly skewed nonsmooth grids. The fourth set of calculations demonstrates the effectiveness of our preconditioner as a function of mesh skewness.

There are two types of meshes used in all four sets of calculations: randomized and Kershaw-squared. Every mesh geometrically models a unit cube, and the outer surface of each mesh conforms exactly to the outer surface of that cube.

Each randomized mesh is generated from an orthogonal mesh composed of uniform cubic cells having a characteristic length, l_c. In particular, each orthogonal-mesh vertex is randomly and uniformly relocated within a sphere of radius r_0, where $r_0 = 0.25\, l_c$, that is centered about the vertex. These randomized meshes are both nonsmooth and skewed, but these properties are approximately constant independent of the mesh size. These meshes are intended to be representative of nonorthogonal meshes that are nonsmooth and skewed but relatively well-behaved. Any scheme that performs well on such meshes should certainly be expected to perform well on smooth meshes. The exterior of the randomized meshes are orthogonal because only the interior mesh points are randomized. The interior of a cubic randomized mesh is illustrated in Fig. 7.

The Kershaw-squared meshes are a 3-D variation on the 2-D Kershaw meshes that first appeared in [12]. The exteriors of a $10 \times 10 \times 10$ Kershaw-squared mesh and a $20 \times 20 \times 20$ Kershaw-squared mesh are illustrated in Figs. 8 and 9, respectively. The interior of a
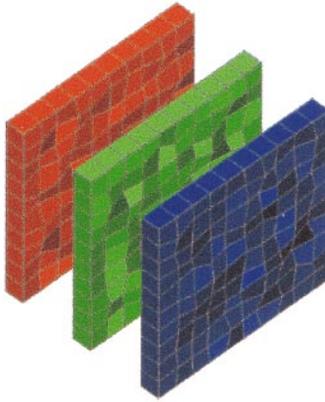
FIG. 7.   Interior view of a portion of a $10 \times 10 \times 10$ random mesh.

$10 \times 10 \times 10$ Kershaw-squared mesh illustrated in Fig. 10. By comparing Figs. 8 and 9, it can be seen that the Kershaw-squared meshes become increasingly nonsmooth and skewed as the mesh size (the number of mesh cells) is increased. Thus, they are representative of severely distorted and highly ill-behaved meshes, and they provide a severe test for hexahedral-mesh discretization schemes.

We have performed test calculations using 3-D meshes for problems with either a 1-D dependence or a full 3-D dependence. We make use of 1-D problems simply because analytic benchmarks are far easier to obtain in 1-D and have often been used in the past to test multidimensional diffusion discretizations [2, 4, 10, 12]. It is important to recognize that the 3-D accuracy of our discretization is in fact being tested in problems with a 1-D dependence because the randomized and Kershaw-squared meshes do not reflect the 1-D symmetry of the solution.

We have computationally compared our support-operators method with another diffu-sion discretization method, which we call the MH method [13]. This method represents a generalization of the 2-D MDHW method [14] to 3-D unstructured hexahedral meshes. The MH method is very similar to our support-operators method. It has the same set of discrete unknowns, and the same discretization for the balance equation. The MH method differs
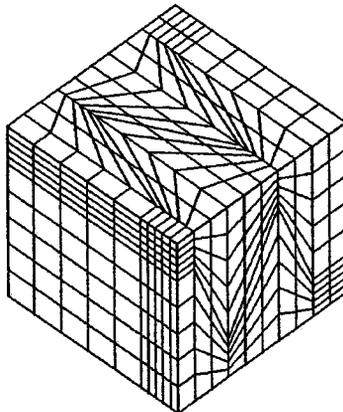
FIG. 8.   External view of a $10 \times 10 \times 10$ Kershaw-squared mesh.
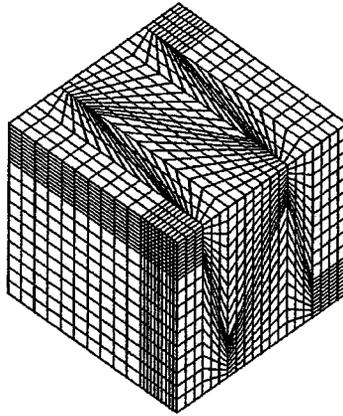
FIG. 9. External view of a $20 \times 20 \times 20$ Kershaw-squared mesh.

from our method only in the computation of the fluxes from the intensities. The main advantage of the MH scheme relative to our support-operators scheme is that it exactly preserves linear homogeneous solutions. This is a direct consequence of the fact that the MH flux expression is exact for a linear intensity dependence. Another slight advantage is that the MH face-center intensity equations have an 11-point stencil rather than the 13-point stencil of our method. The main disadvantage of the MH method relative to our method is that it generates a nonsymmetric coefficient matrix.

The problem associated with the first set of calculations can be described as

$$-D(z)\frac{\partial \phi}{\partial z} = Qz^2, \tag{65}$$

for $z \in [0, 1]$, where

$$D(z) = \begin{cases} D_1, & \text{for } z \in [0, 0.5], \\ D_2, & \text{for } z \in [0.5, 1], \end{cases} \tag{66}$$

with a reflective boundary condition at $z = 0$, a Marshak vacuum boundary condition at $z = 1$, and where $D_1 = \frac{1}{30}, D_2 = \frac{1}{3}$, and $Q = 1$. We refer to this problem as the two-material
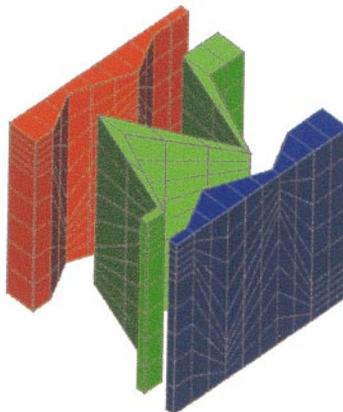
FIG. 10. Interior view of a portion of a $10 \times 10 \times 10$ Kershaw-squared mesh.

problem. This problem has a material discontinuity at $z = 0.5$. The exact solution to the two-material problem is

$$\phi = \begin{cases} a + b + c_1 z^4, & \text{for } z \in [0, 0.5], \\ a + c_2 z^4, & \text{for } z \in [0.5, 1.0], \end{cases} \tag{67}$$

where

$$a = \frac{Q(1 + 8D_2)}{12D_2}, \quad b = \frac{Q(D_2 - D_1)}{192D_1 D_2}, \quad c_1 = -\frac{Q}{12D_1}, \quad c_2 = -\frac{Q}{12D_2}. \tag{68}$$

This problem is solved in 3-D on a unit cube having the vacuum boundary condition on one side of the cube together with reflecting conditions on the remaining five sides.

We have performed a set of calculations for the two-material problem using both our support-operators method and the MH method with grids of the following sizes: $4 \times 4 \times 4$, $8 \times 8 \times 8$, $16 \times 16 \times 16$, $32 \times 32 \times 32$, $45 \times 46 \times 46$, and $54 \times 54 \times 54$. The material discontinuity was only approximately represented on these meshes because all vertices on the mesh interior were randomized. The relative L_2 intensity error was computed for each calculation. This error is defined as the L_2 norm of the difference between the vector of exact cell-center intensities and the vector of computed cell-center intensities divided by the L_2 norm of the vector of exact cell-center intensities, i.e. $\|\hat{\phi}_{exact} - \hat{\phi}_{computed}\|_2 / \|\hat{\phi}_{exact}\|_2$. The errors are plotted as a function of average cell length in Fig. 11 for our support-operators method together with a linear fit to the logarithm of the error as a function of the logarithm of the average cell length. The slope of this linear fit is 1.98. Perfect second-order convergence corresponds to a slope of 2.0. Thus, our support-operators diffusion scheme converges with second-order accuracy for the two-material problem on randomized meshes. The errors for the MH scheme are given in Fig. 12 together with a linear fit. The slope of the fit is
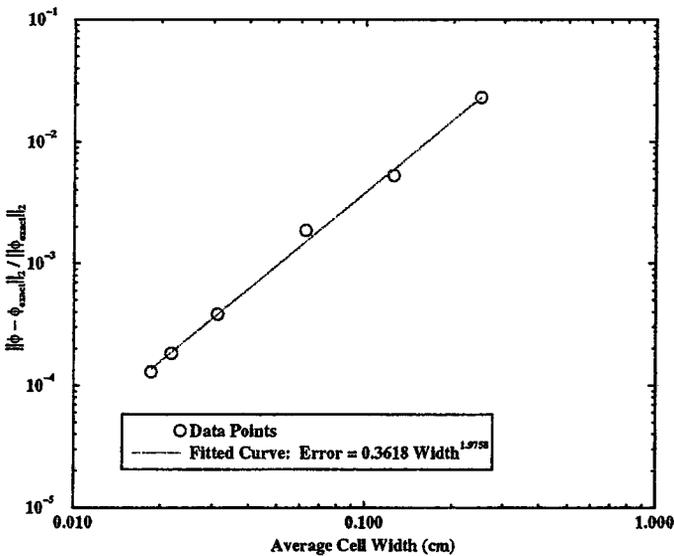
FIG. 11. Convergence data and least-squares linear fit for the support-operators method and the two-material problem.
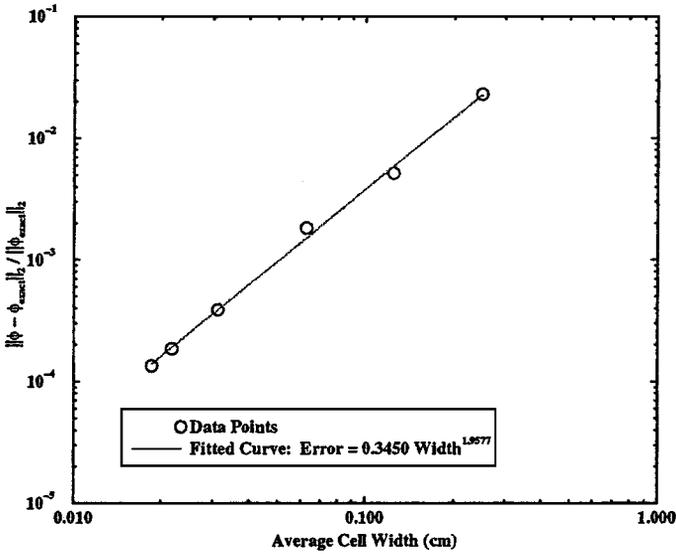
FIG. 12. Convergence data and least-squares linear fit for the MH method and the two-material problem.

1.96. Thus, the MH method also converges with second-order accuracy for the two-material problem on randomized meshes.

In order to determine the sensitivity of the support-operators method to large jumps in the diffusion coefficient, we performed an additional set of calcultions for this problem with $D_1 = \frac{1}{3 \times 10^6}$ rather than $D_1 = \frac{1}{30}$. The errors for the support-operators scheme are given in Fig. 13 together with a linear fit. The slope of the fit is 1.96. Thus, second-order convergence is maintained even with a large jump in the diffusion coefficient. It can be seen by comparing Figs. 11 and 13 that the errors are larger with a larger jump in the diffusion
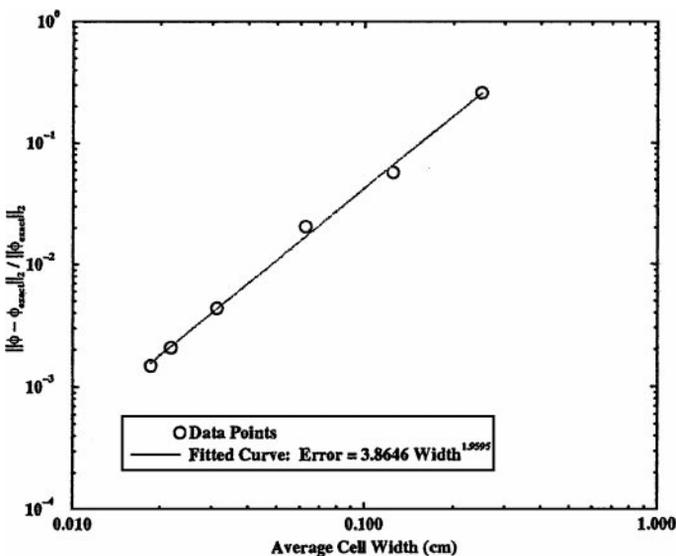
FIG. 13. Convergence data and least-squares linear fit for the support-operators method and the two-material problem with a large coefficient jump.
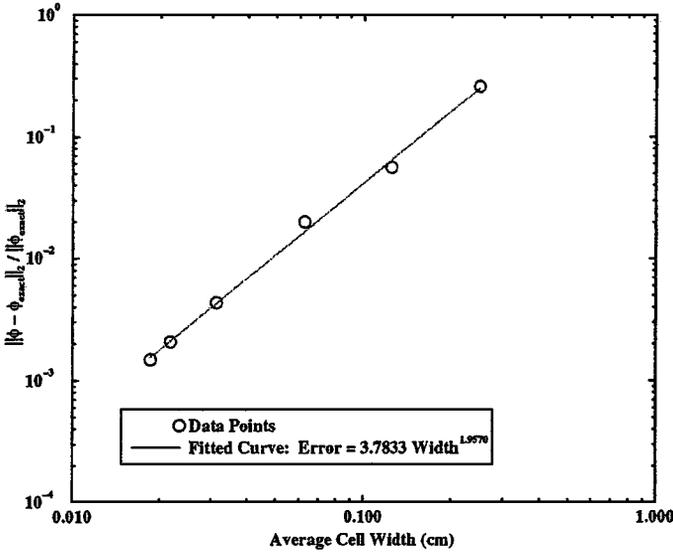
FIG. 14. Convergence data and least-squares linear fit for the MH method and the two-material problem with a large coefficient jump.

coefficient. The errors for the MH scheme are given in Fig. 14 together with a linear fit. The slope of the fit is 1.96. Thus, the MH method also maintains second-order accuracy with a large coefficient jump. A comparison of Figs. 12 and 14 shows a similar increase in the error with a larger jump in the diffusion coefficient.

The problem associated with the third set of calculations has full 3-D dependence and can be described as

$$-\vec{\nabla} \cdot D\vec{\nabla}\phi = Q, \tag{69}$$

for $\vec{r} \in [0, 1] \times [0, 1] \times [0, 1]$, with Marshak boundary conditions on every face having the extrapolated intensities

$$\phi^e(0, y, z) = \phi^e(1, y, z) = \frac{Q}{6D}[y(1 - y) + z(1 - z)], \tag{70}$$

$$\phi^e(x, 0, z) = \phi^e(x, 1, z) = \frac{Q}{6D}[x(1 - x) + z(1 - z)], \tag{71}$$

$$\phi^e(x, y, 0) = \phi^e(x, y, 1) = \frac{Q}{6D}[x(1 - x) + y(1 - y)], \tag{72}$$

where $D = \frac{1}{30}$ and $Q = 1$. We refer to this problem as the 3-D problem. The solution to the 3-D problem is

$$\phi = \frac{Q}{3} + \frac{Q}{6D}[x(1 - x) + y(1 - y) + z(1 - z)]. \tag{73}$$

We have performed a set of calculations using both the support-operators and MH methods for the 3-D problem with both randomized and Kershaw-squared meshes of the following sizes: $4 \times 4 \times 4$, $8 \times 8 \times 8$, $16 \times 16 \times 16$, $32 \times 32 \times 32$, $45 \times 45 \times 45$, and $54 \times 54 \times 54$.
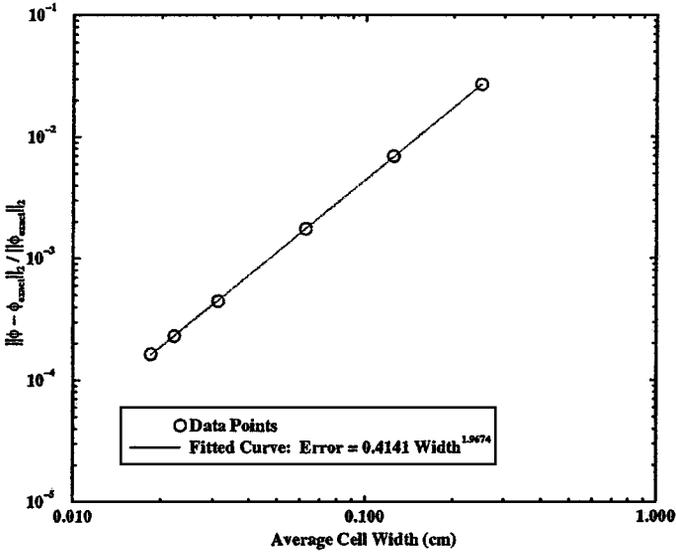
FIG. 15. Convergence data and least-squares linear fit for the support-operators method and the 3-D problem on random meshes.

The relative L_2 intensity error was computed for each calculation. The errors are plotted for the support-operators method on randomized meshes in Fig. 15 together with a linear fit to the data. The slope of the fit is 1.97. Thus, our support-operators method converges with second-order accuracy for this problem. The errors are plotted for the MH method on randomized meshes in Fig. 16 together with a linear fit to the data. The slope of the fit is 2.0. Thus, the MH scheme also converges with second-order accuracy for this problem.

The errors for the 3-D problem are given for both the support-operators method and the MH method on Kershaw-squared meshes in Table I. There is too much noise in the data
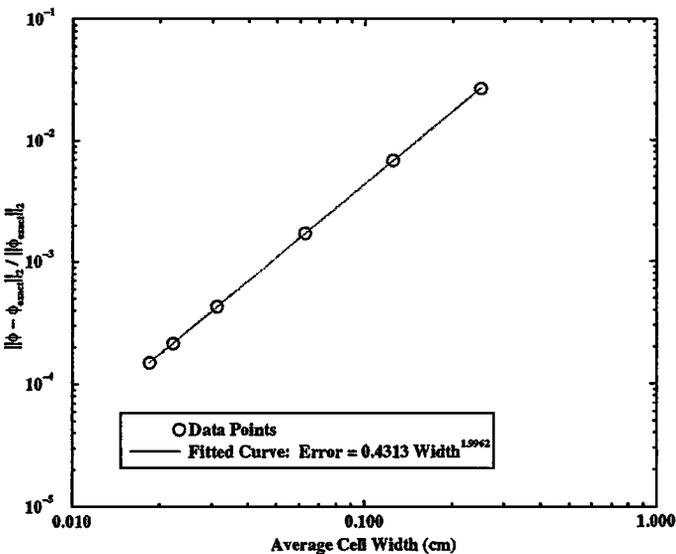
FIG. 16. Convergence data and least-squares linear fit for the MH method and the 3-D problem on random meshes.

TABLE I

**Convergence of the Support-Operators and MH Methods for the 3-D Problem
on Kershaw-Squared Meshes**

Mesh	Scheme	Error	Slope[a]
$4 \times 4 \times 4$	SO	5.61×10^{-2}	—
$8 \times 8 \times 8$	SO	1.85×10^{-2}	1.60
$16 \times 16 \times 16$	SO	7.05×10^{-3}	1.39
$32 \times 32 \times 16$	SO	1.99×10^{-3}	1.82
$45 \times 45 \times 45$	SO	1.10×10^{-3}	1.74
$54 \times 54 \times 54$	SO	8.56×10^{-4}	1.38
$4 \times 4 \times 4$	MH	2.56×10^{-2}	—
$8 \times 8 \times 8$	MH	3.39×10^{-2}	−0.4
$16 \times 16 \times 16$	MH	9.64×10^{-3}	1.81
$32 \times 32 \times 16$	MH	1.56×10^{-3}	2.63
$45 \times 45 \times 45$	MH	7.96×10^{-4}	1.97
$54 \times 54 \times 54$	MH	6.20×10^{-4}	1.37

[a] This column contains the slope of a two-point linear fit calculated using only the data for the mesh assigned the slope value and the data for the next-smallest mesh. For instance, the slope given for the $16 \times 16 \times 16$ mesh was calculated using the data from the calculations for the $16 \times 16 \times 16$ mesh and the $8 \times 8 \times 8$ mesh.

to do a reliable linear fit for either scheme. It is clear from the data in Table I that both the support-operators and MH schemes are converging at a rate faster than first-order, but slower than second-order. Thus, the convergence of both schemes is degraded on Kershaw-squared meshes relative to the convergence obtained on randomized meshes. This is not surprising since the Kershaw-squared meshes are extremely skewed and grow increasingly skewed as the mesh is refined, whereas the randomized meshes have a relatively fixed level of skewing.

The problem associated with the fourth set of calculations can be described as

$$-D\frac{\partial \phi}{\partial z} = Qz^2, \tag{74}$$

for $z \in [0, 1]$, with Marshak vacuum boundary conditions at $z = 0$ and $z = 1$, and where $D = \frac{1}{30}$, and $Q = 1$. We refer to this problem as the homogeneous problem. The homogeneous problem is solved in 3-D on a unit cube by having the vacuum boundary conditions on two opposing sides of the cube with reflecting conditions on the remaining four sides. We have performed calculations for this problem using both random and Kershaw-squared meshes in conjunction with two different solution techniques. The first is to apply row and column scaling to the coefficient matrix and then to solve the resulting system using the conjugate-gradient method in conjunction with symmetric successive overrelaxation (SSOR) for preconditioning. We refer to this as the one-level solution technique. The second is to apply row and column scaling to the coefficient matrix and then solve the resulting system using the conjugate-gradient method in conjunction with the low-order 7-point cell-center diffusion scheme for preconditioning. We refer to this as the two-level solution technique. The low-order equations are solved by first applying row and column scaling to the low-order coefficient matrix and then using the conjugate-gradient method in conjunction with SSOR preconditioning. Note that the low-order system is solved once per

TABLE II
Comparison of One-Level and Two-Level Solution Techniques

Technique	Mesh type	FS iterations	Max LO iterations	CPU time (sec)
One-Level	Random	97	—	143.24
Two-Level	Random	7	32	61.53
One-Level	Kershaw[2]	175	—	247.17
Two-Level	Kershaw[2]	46	42	352.91

full-system conjugate-gradient iteration. The total conjugate-gradient iterations required for the full system, the maximum iterations required for the low-order system, and the total CPU time are given for each calculation in Table II. It can be seen from Table II that the two-level solution technique takes 14 times fewer full-system iterations than the one-level solution technique on the random mesh, but it takes only about 3.5 times fewer full-system iterations on the Kershaw-squared mesh. This is expected since the low-order scheme becomes increasingly inaccurate relative to the full scheme as the mesh becomes increasingly skewed. Note that the two-level scheme is faster than the one-level scheme on the random mesh, but it is slower than the one-level scheme on the Kershaw-squared mesh. The decrease in CPU times for the two-level scheme will be very dependent upon the method used to solve the low-order system. For instance, rather than solve the low-order system to a high level of precision using a Krylov method, one might simply perform a fixed number of multigrid V-cycles. This would greatly reduce the cost of the preconditioning step and thereby reduce the total CPU time as well. Such a strategy was employed with great benefit in [2]. It is important to realize that the structure of the low-order cell-center system on structured meshes is compatible with standard multigrid methods such as Dendy's method [15], whereas the full system has a structure that is incompatible with standard methods. Thus, the low-order preconditioning approach enables highly efficient solution techniques to be used in an indirect manner when they cannot be directly applied to the full system.

6. SUMMARY AND FUTURE WORK

We have developed a cell-centered support-operators diffusion discretization for unstructured hexahedral meshes with spatially-discontinuous diffusion coefficients that produces a sparse symmetric positive-definite coefficient matrix and yields second-order convergence on nonsmooth randomized meshes. We believe that second-order convergence on nonsmooth randomized meshes implies second-order convergence on any type of "well-behaved" nonsmooth meshes, but without a rigorous mathematical proof of convergence, "well-behaved" must remain a subjective concept. Our scheme properly treats material discontinuities in that the normal component of the flux is continuous across such discontinuities and the transverse component may be discontinuous. The main disadvantage of our method is the need for face-center intensities in addition to cell-center intensities. This disadvantage is mitigated by the use of a low-order diffusion discretization as a preconditioner that is symmetric positive-definite and monotone, and has only cell-center intensities in the coefficient matrix. Our support-operators discretization is very similar to hybrid mixed finite-element diffusion discretizations. However, our approach does not require the use of basis functions. Most importantly, current hybrid mixed finite-element methods appear to

require a certain degree of mesh smoothness to be convergent. For these reasons, we feel that our method represents a valuable and unique alternative to existing diffusion discretization schemes for nonorthogonal hexahedral meshes.

There are several possibilities for improving our method. One could certainly reduce the size of the coefficient matrix by locally eliminating the cell-center unknowns. However, this elimination must be done computationally since the matrix elements for our method have to be obtained computationally. Our method would probably best benefit from an improved preconditioner for highly skewed meshes. There are several candidate schemes that we intend to investigate in the future.

APPENDIX

The purpose of this appendix is to demonstrate that the coefficient matrix for our support-operators method is symmetric positive-definite (SPD). This is achieved in the following manner. First, we demonstrate that the $\mathbf{W}$ matrix is SPD. Next, we show that the coefficient matrix for a single-cell problem with reflective boundary conditions is symmetric positive-semidefinite (SPS) with a one-dimensional null space consisting of any set of spatially constant intensities. At this point, the demonstration becomes perfectly analogous to that given in [2] for the 2-D case. We conclude the 3-D demonstration by giving a brief description of the final steps. The full details of these steps are given in [2].

The following mathematical preliminaries are discussed in [9]. A matrix, $\mathbf{B}$, is symmetric if and only if

$$\mathbf{B} = \mathbf{B}^t. \tag{A.1}$$

A matrix, $\mathbf{B}$, is SPD if and only if it is symmetric and it satisfies

$$\hat{X}^t \mathbf{B} \hat{X} > 0, \quad \text{for all vectors } \hat{X}. \tag{A.2}$$

A matrix, $\mathbf{B}$, is SPS if and only if it is symmetric and it satisfies

$$\hat{X}^t \mathbf{B} \hat{X} \geq 0, \quad \text{for all vectors } \hat{X}. \tag{A.3}$$

Thus, every SPD matrix is also SPS. Assume that a square matrix, $\mathbf{B}$, can be expressed in terms of a square matrix, $\mathbf{K}$, as

$$\mathbf{B} = \mathbf{K}^t \mathbf{K}. \tag{A.4}$$

Then if $\mathbf{K}$ is not invertible, $\mathbf{B}$ is SPS but not SPD, and if $\mathbf{K}$ is invertible, $\mathbf{B}$ is SPD.

We begin the overall demonstration by showing that the matrix given in Eq. (31), $\mathbf{W}$, is SPD. It suffices to show that its inverse, explicitly given by Eq. (27) and its analogues, is SPD. We begin the construction of $\mathbf{W}^{-1}$ by considering Eq. (26) and the S-matrices that appear in it. Each of the S-matrices is a 3×3 matrix that is uniquely associated with a vertex, and each of these matrices operates on a 3-vector composed of the face-area flux components associated with that vertex. We now re-express these 3×3 matrices as 6×6 matrices by having them operate on a vector composed of all six face-area flux components

associated with the cell. For instance, the matrix $\mathbf{S}^{LBD}$ operates on the following vertex face-area flux vector:

$$\hat{F}^{LBD} = (f^L, f^B, f^D)^t. \tag{A.5}$$

We want to redefine $\mathbf{S}^{LBD}$ so that it operates on the global vector of flux components:

$$\hat{\mathcal{F}} = (f^L, f^R, f^B, f^T, f^D, f^U)^t. \tag{A.6}$$

This is easily accomplished via a 3×6 matrix that we denote as $\mathbf{P}^{LBD}$. In particular, the 6×6 version of $\mathbf{S}^{LBD}$ is given by

$$\mathbf{S}^{LBD}_{6\times6} = \mathbf{P}^{LBD^t} \mathbf{S}^{LBD} \mathbf{P}^{LBD}, \tag{A.7}$$

where

$$\mathbf{P}^{LBD}_{L,L} = \mathbf{P}^{LBD}_{B,B} = \mathbf{P}^{LBD}_{D,D} = 1, \tag{A.8}$$

and all other elements of $\mathbf{P}^{LBD}$ are zero. The matrix $\mathbf{S}^{LBD}_{6\times6}$ is explicitly given by

$$\mathbf{P}^{LBD^t} \mathbf{S}^{LBD} \mathbf{P}^{LBD} = \begin{bmatrix} s_{L,L} & 0 & s_{L,B} & 0 & s_{L,D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ s_{B,L} & 0 & s_{B,B} & 0 & s_{B,D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ s_{D,L} & 0 & s_{D,B} & 0 & s_{D,D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{A.9}$$

For the general case, the matrix $\mathbf{P}$ is most easily defined with respect to the matrix $\mathbf{S}$ using numeric indices. To do this we simply number all vector components in the usual sequential manner, e.g.,

$$(f^L, f^B, f^D)^t \rightarrow (f_1, f_2, f_3)^t, \tag{A.10}$$

and

$$(f^L, f^R, f^B, f^T, f^D, f^U)^t \rightarrow (f_1, f_2, f_3, f_4, f_5, f_6)^t. \tag{A.11}$$

Using this numeric indexing, the matrix $\mathbf{P}$ is defined for the general case as follows: If the i'th component of the local vector $\hat{F}^{vertex}$ associated with $\mathbf{S}^{vertex}$ is the j'th component of the global vector $\hat{\mathcal{F}}$, then

$$p_{i,j} = 1, \tag{A.12}$$

otherwise

$$p_{i,j} = 0. \tag{A.13}$$

It is convenient at this point to assign the vertices with the indices LBD, RBD, LTD, RTD, LBU, RBU, LTU, RTU, to the respective numeric indices 1, 2, 3, 4, 5, 6, 7, 8. This enables us to re-express Eq. (26) as

$$\hat{\mathcal{H}}^t \hat{\Phi} + D^{-1} \sum_{n=1}^{8} V_n \hat{\mathcal{H}}^t \mathbf{P}_n^t \mathbf{S}_n \mathbf{P}_n \hat{\mathcal{F}} = \hat{\mathcal{H}}^t (\phi^C \hat{1}), \tag{A.14}$$

where n is the numeric vertex index, and where

$$\hat{1} = (1, 1, 1, 1, 1, 1)^t, \tag{A.15}$$

$$\hat{\Phi} = (\phi^L, \phi^R, \phi^B, \phi^T, \phi^D, \phi^U)^t, \tag{A.16}$$

$$\hat{\mathcal{H}} = (h^L, h^R, h^B, h^T, h^D, h^U)^t. \tag{A.17}$$

Since Eq. (A.14) must hold for all possible $\hat{\mathcal{H}}$, it follows that

$$\hat{\Phi} + D^{-1} \left[\sum_{n=1}^{8} V_n \mathbf{P}_n^t \mathbf{S}_n \mathbf{P}_n \right] \hat{\mathcal{F}} = \phi_C \hat{1}. \tag{A.18}$$

Further manipulating Eq. (A.18), we obtain

$$D^{-1} \left[\sum_{n=1}^{8} V_n \mathbf{P}_n^t \mathbf{S}_n \mathbf{P}_n \right] \hat{\mathcal{F}} = \Delta \hat{\Phi}, \tag{A.19}$$

where $\Delta \hat{\Phi}$ is defined by Eq. (30). Comparing Eqs. (28) and (A.19) it follows that

$$\mathbf{W}^{-1} = D^{-1} \left[\sum_{n=1}^{8} V_n \mathbf{P}_n^t \mathbf{S}_n \mathbf{P}_n \right]. \tag{A.20}$$

From Eq. (20) it follows that each 3×3 **S**-matrix is the product of a matrix **A** and its transpose. Substituting from Eq. (20) into Eq. (A.20), we get,

$$\mathbf{W}^{-1} = D^{-1} \left[\sum_{n=1}^{8} V_n \mathbf{P}_n^t \mathbf{A}_n^t \mathbf{A}_n \mathbf{P}_n \right],$$

$$= D^{-1} \left[\sum_{n=1}^{8} V_n (\mathbf{A}_n \mathbf{P}_n)^t (\mathbf{A}_n \mathbf{P}_n) \right]. \tag{A.21}$$

Since

- the matrix, $(\mathbf{A}_n \mathbf{P}_n)^t (\mathbf{A}_n \mathbf{P}_n)$, must be SPS for each value of n,
- an SPS matrix multiplied by a positive scalar remains SPS,
- the diffusion coefficient will always be positive,
- the vertex volumes will be positive with any reasonably well-formed mesh,
- the **A**-matrices will be invertible with any well-formed mesh,
- the **P**-matrices are not invertible,

it follows from Eq. (A.21) that $\mathbf{M}_n$ must be SPS but not SPD for each value of n, where

$$\mathbf{M}_n = D^{-1}V_n(\mathbf{A}_n\mathbf{P}_n)^t(\mathbf{A}_n\mathbf{P}_n). \tag{A.22}$$

Substituting from Eq. (A.22) into Eq. (A.21) we find that $\mathbf{W}^{-1}$ is a sum of matrices with each constituent matrix, $\mathbf{M}_n$, being SPS:

$$\mathbf{W}^{-1} = \sum_{n=1}^{8} \mathbf{M}_n. \tag{A.23}$$

It is shown in [2] that if a matrix is a sum of SPS matrices, it is SPS, and its null space is the intersection of the null spaces of the constituent matrices. From the definitions of the $\mathbf{A}$-matrices and the $\mathbf{P}$-matrices (see Eqs. (18), (A.12), and (A.13)), it follows that each $\mathbf{M}$-matrix has a three-dimensional null space. For instance, the null space of $\mathbf{M}_1$ (corresponding to the LBD corner) consists of any vector of the form

$$\hat{\mathcal{F}} = (0, f^R, 0, f^T, 0, f^U)^t, \tag{A.24}$$

where f^R, f^T, and f^U are free to take on any values. There is no one face-area flux component that is common to the null spaces of all eight $\mathbf{M}$-matrices, so the intersection of their null spaces is the null set. This implies that $\mathbf{W}^{-1}$ has an empty null space. Since it is also SPS, it follows that $\mathbf{W}^{-1}$ is SPD. Finally, if $\mathbf{W}^{-1}$ is SPD, then $\mathbf{W}$ must be SPD.

The next step in the demonstration is to construct the discrete diffusion equations for a single cell with reflective boundary conditions. Let us assume a solution vector, $\hat{\Phi}$, of the form given in Eq. (A.16). In order to use numeric indices for the coefficient matrix of the single-cell system, we number this vector in the usual manner, i.e.,

$$(\phi^L, \phi^R, \phi^B, \phi^T, \phi^D, \phi^U, \phi^C)^t \rightarrow (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7)^t. \tag{A.25}$$

The first six equations for a single cell are the equations for the face-center intensities. For a reflective boundary condition, these equations simply state that the face-area flux component on each face is zero. However, in analogy with Eqs. (41) through (43), we equivalently require that the *negative* of each component be zero. The $\mathbf{W}$-matrix relates the face-area flux components to the differences between the cell-center intensity and the face-center intensities in accordance with Eq. (31). Thus the first six equations can be expressed in terms of the matrix $\mathbf{W}$ as

$$-\mathbf{W}\Delta\hat{\Phi} = 0, \tag{A.26}$$

where in accordance with Eqs. (30) and (A.25),

$$\Delta\hat{\Phi} = (\phi_7 - \phi_1, \phi_7 - \phi_2, \phi_7 - \phi_3, \phi_7 - \phi_4, \phi_7 - \phi_5, \phi_7 - \phi_6)^t. \tag{A.27}$$

Using Eqs. (A.26), and (A.27), one can easily construct the first six rows of the single-cell coefficient matrix, $\mathbf{C}$, as follows:

$$c_{i,j} = \mathbf{W}_{i,j}, \quad i = 1, 6, \quad j = 1, 6, \tag{A.28}$$

$$c_{i,7} = -\sum_{j=1}^{6} \mathbf{W}_{i,j}, \quad i = 1, 6. \tag{A.29}$$

The seventh and last row of $\mathbf{C}$ corresponds to Eq. (34), the steady-state balance equation. Using Eqs. (31), (34), and (A.27) through (A.29), we define the last row of the coefficient matrix:

$$c_{7,j} = -\sum_{i=1}^{6} \mathbf{W}_{i,j}, \quad i = 1, 6 \tag{A.30}$$

$$c_{7,7} = \sum_{i=1}^{6}\sum_{j=1}^{6} \mathbf{W}_{i,j}. \tag{A.31}$$

To summarize, the coefficient matrix takes the block form

$$\mathbf{C} = \begin{bmatrix} \mathbf{W} & \mathbf{W}_r \\ \mathbf{W}_c & \mathbf{W}_{rc} \end{bmatrix}, \tag{A.32}$$

where $\mathbf{W}_r$ is a 6×1 matrix obtained by summing the rows of $\mathbf{W}$, $\mathbf{W}_c$ is a 1×6 matrix obtained by summing the columns of $\mathbf{W}$, and $\mathbf{W}_{rc}$ is a 1×1 matrix obtained by summing all of the elements of $\mathbf{W}$. Note that $\mathbf{W}_c$ is the transpose of $\mathbf{W}_r$ because $\mathbf{W}$ is symmetric. Thus, $\mathbf{C}$ is symmetric. To prove that it is SPS, we need only show that it is positive-semidefinite. Toward this end, we note that any vector $\hat{\Phi}$ can clearly be re-expressed as

$$\hat{\Phi} = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7)^t = \hat{\Phi}_f + \hat{\Phi}_c, \tag{A.33}$$

where

$$\hat{\Phi}_f = (\phi_1 - \phi_7, \phi_2 - \phi_7, \phi_3 - \phi_7, \phi_4 - \phi_7, \phi_5 - \phi_7, \phi_6 - \phi_7, 0)^t, \tag{A.34}$$

and

$$\hat{\Phi}_c = (\phi_7, \phi_7, \phi_7, \phi_7, \phi_7, \phi_7, \phi_7)^t. \tag{A.35}$$

Taking the inner product of $\hat{\Phi}$ with $\mathbf{C}\hat{\Phi}$, we get

$$(\hat{\Phi}_f + \hat{\Phi}_c)^t \mathbf{C}(\hat{\Phi}_f + \hat{\Phi}_c) = \hat{\Phi}_f^t \mathbf{C}\hat{\Phi}_f + \hat{\Phi}_f^t \mathbf{C}\hat{\Phi}_c + \hat{\Phi}_c^t \mathbf{C}\hat{\Phi}_f + \hat{\Phi}_c^t \mathbf{C}\hat{\Phi}_c. \tag{A.36}$$

It is easily verified that

$$\mathbf{C}\hat{\Phi}_c = \hat{0}, \quad \text{for all } \hat{\Phi}_c. \tag{A.37}$$

Substituting from Eq. (A.37) into Eq. (A.36), we get

$$(\hat{\Phi}_f + \hat{\Phi}_c)^t \mathbf{C}(\hat{\Phi}_f + \hat{\Phi}_c) = \hat{\Phi}_f^t \mathbf{C}\hat{\Phi}_f + \hat{\Phi}_c^t \mathbf{C}\hat{\Phi}_f. \tag{A.38}$$

Since

$$\hat{\Phi}_c^t \mathbf{C}\hat{\Phi}_f = \hat{\Phi}_f^t \mathbf{C}^t \hat{\Phi}_c = 0, \tag{A.39}$$

Eq. (A.38) reduces to

$$(\hat{\Phi}_f + \hat{\Phi}_c)^t \mathbf{C}(\hat{\Phi}_f + \hat{\Phi}_c) = \hat{\Phi}_f^t \mathbf{C}\hat{\Phi}_f. \tag{A.40}$$

Using Eq. (A.32), it is easily shown that

$$\hat{\Phi}_f^t \mathbf{C}\hat{\Phi}_f = \hat{\Phi}_{f6}^t \mathbf{W}\hat{\Phi}_{f6}, \tag{A.41}$$

where

$$\hat{\Phi}_{f6} = (\phi_1 - \phi_7, \phi_2 - \phi_7, \phi_3 - \phi_7, \phi_4 - \phi_7, \phi_5 - \phi_7, \phi_6 - \phi_7)^t. \tag{A.42}$$

Since $\mathbf{W}$ is SPD, it follows from Eqs. (A.40) through (A.42) that

$$(\hat{\Phi}_f + \hat{\Phi}_c)^t \mathbf{C}(\hat{\Phi}_f + \hat{\Phi}_c) = 0, \quad \text{if } \hat{\Phi}_f = \hat{0},$$
$$> 0, \quad \text{otherwise.} \tag{A.43}$$

Thus, $\mathbf{C}$ is positive-semidefinite. Since it is also symmetric, $\mathbf{C}$ is SPS. Note from Eq. (A.43) that the null space of $\mathbf{C}$ is spanned by all vectors $\hat{\phi}_c$. Following Eq. (A.35), it is clear that the null space of $\mathbf{C}$ is spanned by all vectors of constant intensity.

The remainder of the demonstration is identical to that given for the 2-D case in [2]. The final steps can be briefly described as follows:

1. Given a multicell mesh with N cells, the $\mathbf{C}$-matrices for each cell are expanded to operate on the global vector of intensities for the entire mesh. This step is conceptually analogous to the expansion of the $\mathbf{S}^{LBD}$ matrix given in Eq. (A.9). Since the $\mathbf{C}$-matrices are SPS, their expansions must be SPS.

2. It is shown that the sum of the expanded $\mathbf{C}$-matrices represents the coefficient matrix for the entire mesh with reflective conditions on the outer boundary faces. Since the global coefficient matrix is the sum of SPS matrices, it must be SPS. Furthermore, the null space of the full coefficient matrix must be equal to the intersection of the null spaces of the expanded $\mathbf{C}$-matrices.

3. It is shown that the null space of the full coefficient matrix is spanned by all vectors of constant intensity. This is the correct result because the analytic diffusion operator has a null space spanned by all constant intensity functions if the reflective condition is imposed on the entire outer boundary. The analytic diffusion operator becomes invertible if the reflective condition is replaced with an extrapolated boundary condition on any portion of the outer boundary surface.

4. Finally, it is shown that if the reflective boundary condition is replaced with an extrapolated condition on any outer-boundary cell face, the expanded $\mathbf{C}$-matrix for the cell containing the boundary face has a null space that is disjoint from the null spaces of all the other expanded $\mathbf{C}$-matrices. Thus, the intersection of the null spaces of all the expanded $\mathbf{C}$-matrices is the null set. Since the global coefficient matrix is the sum of the expanded $\mathbf{C}$-matrices, and the expanded $\mathbf{C}$-matrices are SPS, it follows that the global coefficient matrix is SPD.

REFERENCES

1. O. C. Zienkiewicz, *The Finite Element Method*, 3rd ed. (McGraw-Hill, London, 1977).

2. J. E. Morel, Randy M. Roberts, and Mikhail J. Shashkov, A Local Support-Operators Diffusion Discretization Scheme for Quadrilateral *r-z* Meshes, *J. Comput. Phys.* **144**, 17 (1998).

3. G. C. Pomraning, *Equations of Radiation Hydrodynamics*, International Series of Monographs in Natural Philosophy, edited by D. ter Haar (Pergamon, Elmsford, NY, 1973), Vol. 54.

4. A. I. Shestakov, J. A. Harte, and D. S. Kershaw, Solution of the diffusion equation by finite elements in Lagrangian hydrodynamics codes, *J. Comput. Phys.* **76**, 385 (1988).

5. Milton E. Rose, Compact volume methods for the diffusion equation, *J. Sci. Comput.* **4**, 261 (1989).

6. Todd Arbogast, Clint N. Dawson, Philip T. Keenan, Mary F. Wheeler, and Ivan Yotov, Enhanced cell-centered finite differences for elliptic equations on general geometry, *SIAM J. Sci. Comput.* **18**, 1 (1997).

7. James Hyman, Mikhail Shashkov, and Stanly Steinberg, *Inner Products for Discrete Vector Fields and Accuracy of Mimetic Finite Difference Methods*, Los Alamos National Report LA-UR-99-6181 (1999).

8. A. Weiser and M. F. Wheeler, On convergence of block-centered finite differences for elliptic problems, *SIAM J. Numer. Anal.* **25**, 351 (1988).

9. Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, 2nd ed. (Johns Hopkins Press, Baltimore, 1989).

10. M. J. Shashkov and S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, *J. Comput. Phys.* **129**, 383 (1996).

11. J. D. Jackson, *Classical Electrodynamics*, 2nd ed. (Wiley, New York, 1975).

12. D. S. Kershaw, Differencing of the diffusion equation in Lagrangian hydrodynamics codes, *J. Comput. Phys.* **39**, 375 (1981).

13. Michael L. Hall and Jim E. Morel, *A Second-Order Cell-Centered Diffusion Scheme for Unstructured Hexahedral Meshes*, Los Alamos National Laboratory Report LA-CP-97-8 (1997).

14. J. E. Morel, J. E. Dendy, Jr., Michael L. Hall, and Stephen W. White, A cell-centered Lagrangian-mesh diffusion differencing scheme, *J. Comput. Phys.* **103**, 286 (1992).

15. J. E. Dendy, Jr., Two multigrid methods for three-dimensional problems with discontinuous and anisotropic coefficients, *SIAM Sci. Stat. Comput.* **8**, 673 (1987).